

(43)Date of publication of application : **13.10.1998**

H04N 7/30

(22)Date of filing : **31.03.1997** (72)Inventor : **MIYAHARA NOBUSADA**  
**YAGASAKI YOICHI**  
**SUZUKI TERUHIKO**

**SPECIFICATION <EXCERPT>**

[0059]

On the other hand, in the case of the MPEG 1/2 mode, the process proceeds to Step S6, and the setting device 132 firstly sets the quantization matrix of MPEG 1/2 type for each block of DCT coefficients having eight elements in the respective vertical and horizontal directions. The quantization matrix set here differs between the intra picture coding and the inter picture coding. A new matrix may be set and used instead of the quantization matrix prepared beforehand. In this case, a flag of load\_intra\_quant\_mat or load\_nonintra\_quant\_mat of Video Object Layer Class is set, and 64 elements of the newly set quantization matrix are coded as intra\_quant\_mat [64] or nonintra\_quant\_mat [64].

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-276097

(43) 公開日 平成10年(1998)10月13日

(51) Int.Cl.<sup>8</sup>

識別記号

F I

H 0 3 M 7/30

H 0 3 M 7/30

B

H 0 4 N 1/41

H 0 4 N 1/41

B

7/30

7/133

Z

審査請求 未請求 請求項の数6 O L (全 34 頁)

(21) 出願番号 特願平9-79802

(22) 出願日 平成9年(1997)3月31日

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 宮原 信禎

東京都品川区北品川6丁目7番35号 ソニ

ー株式会社内

(72) 発明者 矢ヶ崎 陽一

東京都品川区北品川6丁目7番35号 ソニ

ー株式会社内

(72) 発明者 鈴木 輝彦

東京都品川区北品川6丁目7番35号 ソニ

ー株式会社内

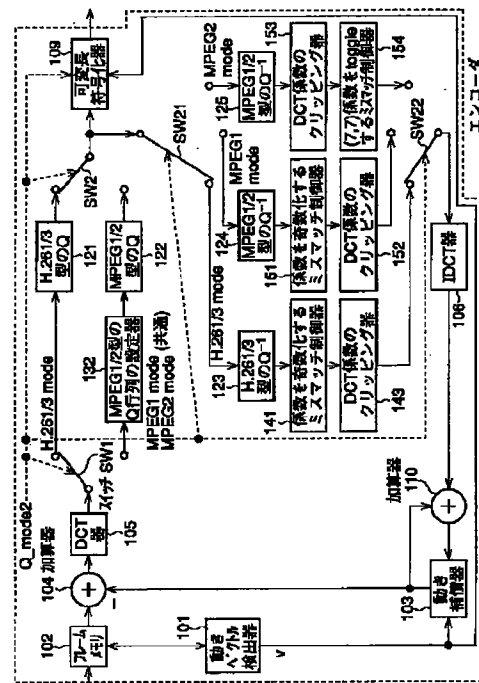
(74) 代理人 弁理士 稲本 義雄

(54) 【発明の名称】 符号化装置および方法、並びに復号装置および方法

(57) 【要約】

【課題】 最適な組み合わせを指定し、高能率の符号化を行うことができるようにする。

【解決手段】 スイッチSW1、SW2により、H. 261/3モードによる量子化、または、設定器132による量子化行列の設定後、量子化器122によるMPEG1/2型の量子化のいずれかが選択され、スイッチSW21、SW22により、H. 261/3モードでの逆量子化器123、ミスマッチ制御器141、クリッピング器143による符号化処理、または、MPEG1モードでの逆量子化器124、ミスマッチ制御器151、クリッピング器152による符号化処理、または、MPEG2モードでの逆量子化器125、クリッピング器153、ミスマッチ制御器154による符号化処理のいずれかが選択され、実行される。



## 【特許請求の範囲】

【請求項1】 データを符号化する符号化装置であって、  
前記データを複数の方法で量子化する量子化手段と、  
前記量子化手段の出力を複数の方法で逆量子化する逆量子化手段と、  
前記量子化手段および前記逆量子化手段で用いる量子化行列を設定する設定手段と、  
前記逆量子化手段の出力に対して複数の方法でIDCTミスマッチ対策を施すミスマッチ対策手段と、  
前記量子化手段が前記データを量子化する前記方法の任意のものと、前記逆量子化手段が前記量子化手段の出力を逆量子化する前記方法の任意のものと、前記ミスマッチ対策手段が前記逆量子化手段の出力に対してIDCTミスマッチ対策を施す前記方法の任意のものと、前記量子化手段および前記逆量子化手段が前記量子化行列を使用する任意の方法とを組み合わせる前記データを符号化する方法を制御する制御手段とを備えることを特徴とする符号化装置。

【請求項2】 データを符号化する符号化方法であって、  
前記データを複数の方法で量子化するステップと、  
前記量子化された前記データを複数の方法で逆量子化するステップと、  
量子化および逆量子化を行うときに用いる量子化行列を設定するステップと、  
逆量子化された前記データに対して複数の方法でIDCTミスマッチ対策を施すステップと、  
前記データを量子化する前記方法の任意のものと、量子化された前記データを逆量子化する前記方法の任意のものと、逆量子化された前記データに対してIDCTミスマッチ対策を施す前記方法の任意のものと、前記データを量子化するときに、および量子化された前記データを逆量子化するときに前記量子化行列を使用する任意の方法とを組み合わせる前記データを符号化する方法を制御するステップとを備えることを特徴とする符号化方法。

【請求項3】 データを復号する復号装置であって、  
前記データを複数の方法で逆量子化する逆量子化手段と、  
前記逆量子化手段で用いる量子化行列を設定する設定手段と、  
前記逆量子化手段の出力に対して複数の方法でIDCTミスマッチ対策を施すミスマッチ対策手段と、  
前記逆量子化手段が前記データを逆量子化する前記方法の任意のものと、前記ミスマッチ対策手段が前記逆量子化手段の出力に対してIDCTミスマッチ対策を施す前記方法の任意のものと、前記逆量子化手段が前記量子化行列を使用する任意の方法とを組み合わせる前記データを復号する方法を制御する制御手段とを備えることを特徴とする復号装置。

【請求項4】 データを復号する復号方法であって、  
前記データを複数の方法で逆量子化するステップと、  
前記データを逆量子化するとき用いる量子化行列を設定するステップと、  
逆量子化された前記データに対して複数の方法でIDCTミスマッチ対策を施すステップと、  
前記データを逆量子化する前記方法の任意のものと、逆量子化された前記データに対してIDCTミスマッチ対策を施す前記方法の任意のものと、前記データを逆量子化するときに前記量子化行列を使用する任意の方法とを組み合わせる前記データを復号する方法を制御するステップとを備えることを特徴とする復号方法。

【請求項5】 データを復号する復号装置であって、  
前記データを複数の方法で逆量子化する逆量子化手段と、  
前記逆量子化手段で用いる量子化行列を設定する設定手段と、  
前記逆量子化手段の出力に対して所定の方法でIDCTミスマッチ対策を施すミスマッチ対策手段と、  
前記逆量子化手段が前記データを逆量子化する前記方法の任意のものと、前記逆量子化手段が前記量子化行列を使用する任意の方法とを組み合わせる前記データを復号する方法を制御する制御手段とを備えることを特徴とする復号装置。

【請求項6】 データを復号する復号方法であって、  
前記データを複数の方法で逆量子化するステップと、  
前記データを逆量子化するとき用いる量子化行列を設定するステップと、  
逆量子化された前記データに対して所定の方法でIDCTミスマッチ対策を施すステップと、  
前記データを逆量子化する前記方法の任意のものと、前記データを逆量子化するときに前記量子化行列を使用する任意の方法とを組み合わせる前記データを復号する方法を制御するステップとを備えることを特徴とする復号方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、符号化装置および方法、並びに復号装置および方法に関し、例えば、複数の量子化および逆量子化の方法、量子化行列の利用方法、IDCTミスマッチの対策方法等を用意し、最適な方法を組み合わせることにより、高能率の符号化および復号を行うようにした符号化装置および方法、並びに復号装置および方法に関する。

## 【0002】

【従来の技術】例えば、テレビ会議システム、テレビ電話システムなどのように、動画像信号を遠隔地に伝送するシステムにおいては、伝送路を効率良く利用するため、映像信号のライン相関やフレーム間相関を利用して、画像信号を圧縮符号化するようになされている。

【0003】動画像の高効率符号化方法の代表的なものとして、MPEG（蓄積用動画像符号化）方式がある。これには、ISO/IEC JTC1/SC29/WG11にて議論され標準案として提案されたMPEG1、MPEG2、現在議論されているMPEG4等があり、主に動き補償予測符号化とDCT（Discrete Cosine Transform）符号化を組み合わせたハイブリッド方式が採用されている。

【0004】以下では、MPEG4のVM6.0（ISO/IEC JTC1/SC29/WG11 MPEG96/N1582Sevilla, February 1996）を例にし、説明を行なう。

【0005】図13、図14を用いて従来の動画像符号化方法、および復号方法について説明する。図13は符号化器を、図14は復号器の構成例を示している。動きベクトル検出器101は、フレームメモリ102に保存\*

$$I_2[i, j, t] = (I[i_2, j_2, t-T] + I[i_2+1, j_2, t-T] + I[i_2, j_2+1, t-T] + I[i_2+1, j_2+1, t-T]) / 4$$

$$i_2 = \text{int}(i + v_x(i, j, t)T)$$

$$j_2 = \text{int}(j + v_y(i, j, t)T)$$

・・・(式1)

【0008】ここで、Tは、現在予測を行っている画像Iが入力された時刻と、フレームメモリ上にある画像が入力された時刻の差であり、式(1)の右辺のI

$I[i_2, j_2, t-T]$ ,  $I[i_2+1, j_2, t-T]$ ,  $I[i_2, j_2+1, t-T]$ ,  $I[i_2+1, j_2+1, t-T]$ は、図示せぬフレームメモリ上の画素値を表わす。また、 $\text{int}(x)$ はxを越えない最大の整数値を表している。

【0009】加算器104は、現在符号化すべき画素の値と、動き補償器103によって算出された予測値との差分を計算する。DCT（Discrete Cosine Transform）器105は、差分値で構成された8画素×8画素のブロックに対して2次元DCTを施す。

【0010】次に、DCT器105によって得られたDCT係数cに対して、適当なステップサイズQを用いて式(2)に象徴される量子化処理を行う。

$$C_2 = \text{int}(C/Q) \quad \dots (式2)$$

【0012】この量子化を行う処理方法は実際には二通り用意されており、フラグQ\_modeと連動したスイッチSW1, SW2によって切り替えられる。フラグQ\_modeは、量子化等を行う際の処理方法の組み合わせを示しており、H.261あるいはH.263（以下、H.261/3と示す）のモード、MPEG1あるいはMPEG2（以下、MPEG1/2と示す）のモードのいずれかが、何らかの方法によってあらかじめ設定されている。H.261/3モードにおいては量子化器121でH.261/3型の量子化が行われる。MPEG1/2モードにおいては、MPEG1/2型の量子化行列がMPEG1/2型のQ行列の設定器（以下では、適宜、設定器と略記する）132で設定された後、量子化器122で

\*された入力画像から、動きベクトルvを検出する。動きベクトルの検出法としては、通常、16画素×16画素のブロック単位でのブロックマッチングが行われる。また、より高い精度を実現するために、半画素単位のマッチングが行われる。

【0006】動き補償器103は図示せぬフレームメモリを備え、現在、符号化すべき画像の各位置の画素値を、すでに符号化/復号が終了し、このフレームメモリに保存されている画像から予測する。時刻tに入力された画像上の位置(i, j)における画素値 $I[i, j, t]$ の予測値 $I_2[i, j, t]$ は、この位置に対応する動きベクトルを $v = (v_x(i, j, t), v_y(i, j, t))$ を用いて、次式のように決定される。

【0007】

MPEG1/2型の量子化が行われる。それぞれの量子化の処理方法および量子化行列の詳細については、後述する。

【0013】量子化の施されたDCT係数は、可変長符号化器109、および逆量子化器123または逆量子化器124に供給される。逆量子化器123, 124でも、量子化器で用いられたのと同じステップサイズにより、次式(3)に象徴される逆量子化処理が行われる。

$$C_3 = C_2 \times Q \quad \dots (式3)$$

【0015】この逆量子化を行う処理方法も実際には二通り用意されており、フラグQ\_modeと連動したスイッチSW3, SW4によって切り替えられる。H.261/3モードにおいては、逆量子化器123でH.261/3型の逆量子化が行われ、ミスマッチ制御器141でDCT係数を奇数化するミスマッチ制御が行われた後、クリッピング器143によってDCT係数がクリッピングされる。MPEG1/2モードにおいては、124の逆量子化器でMPEG1/2型の逆量子化が行われた後、クリッピング器143によってDCT係数がクリッピングされる。それぞれの逆量子化の処理方法およびミスマッチの制御方法の詳細については、後述する。

【0016】逆量子化されクリッピングが施されたデータは、IDCT（Inverse DCT）器106によって逆DCTが行われ、画素値の差分値が復元される。

【0017】この差分値は、加算器110によって動き補償器103から出力される予測値と加算されて画素値のデータとなり、動き補償器103に送られて動き補償器103が備える図示せぬフレームメモリに保存される。

【0018】可変長符号化器109は、量子化器121

あるいは量子化器122によって得られた量子化されたDCT係数、動きベクトル検出器101によって得られた動きベクトル $v$ 、および何らかの方法によって設定されたフラグ $Q\_mode$ に対して可変長符号化処理を施し、符号化ビット列を出力する。

【0019】一方、図14に示した復号器においては、はじめに逆可変長符号化器209によって符号化器(図13)における可変長符号化器109の逆処理が施され、符号化ビット列から、量子化が施されたDCT係数、動きベクトル $v$ 、およびフラグ $Q\_mode$ を復元する。フラグ $Q\_mode$ は連動したスイッチSW11, SW12を作動させ、得られたDCT係数は逆量子化器223、または設定器232を介して逆量子化器224に、動きベクトル $v$ は動き補償器203にそれぞれ供給される。

【0020】逆量子化を行う処理方法も実際には二通り用意されており、フラグ $Q\_mode$ と連動したスイッチSW11, SW12によって切り替えられる。H. 261/3モードにおいては量子化器223でH. 261/3型の逆量子化が行われ、ミスマッチ制御器241でDCT係数を奇数化するミスマッチ制御が行われた後、クリッピング器243によってDCT係数がクリッピングされる。MPEG1/2モードにおいては、MPEG1/2型の量子化行列が設定器232で設定された後、逆量子化器224でMPEG1/2型の逆量子化が行われ、クリッピング器244によってDCT係数がクリッピングされる。なお、双方のモードで用いられる処理器群は、図13に示した符号化器のものと同一のものをを用いることができる。

【0021】また、IDCT器206も図13に示した符号化器のIDCT器106と同一のものであり、IDCTが施されて画素値の差分値が復元される。

【0022】この差分値は、加算器210によって、動き補償器203によってすでに生成されている予測値と

$$\begin{aligned} \text{For INTRA: LEVEL} &= |\text{COF}| / (2 \times \text{QP}) \\ \text{For INTER: LEVEL} &= (|\text{COF}| - \text{QP} / 2) / (2 \times \text{QP}) \\ \text{QCOF} &= \text{sign}(\text{COF}) \times \text{LEVEL} \end{aligned}$$

... (式4)

【0028】H. 261/3型の逆量子化は、次式 ※T係数を表わす。

(5) のように行われる。COF<sub>2</sub>は逆量子化後のDCT

$$\begin{aligned} \text{if LEVEL} = 0: |\text{COF}_2| &= 0 \\ \text{if LEVEL} \neq 0: |\text{COF}_2| &= 2 \times \text{QP} \times \text{LEVEL} + \text{QP} \\ \text{COF}_2 &= \text{sign}(\text{COF}) \times |\text{COF}_2| \end{aligned}$$

... (式5)

【0030】MPEG1/2型の量子化は、例えば図16に示したように、量子化後に0となる範囲が量子化幅と同一であるような量子化である (dead zoneの無い量子化)。  
★子化)。VM6.0では次式(6)のように行われ、定数 $q$ ,  $p$ は現時点で $p=3$ ,  $q=4$ となっている。

$$\begin{aligned} \text{For INTRA: QCOF} &= (\text{COF} + \text{sign}(\text{COF}) \times ((p \times \text{QP}) // q)) / (2 \times \text{QP}) \\ \text{For INTER: QCOF} &= \text{COF} / (2 \times \text{QP}) \end{aligned}$$

... (式6)

【0032】MPEG1/2型の量子化は、次式(7) のように行われる。COF<sub>2</sub>は逆量子化後のDCT係数

動き補償器203によってすでに生成されている予測値と加算されて画素値のデータとなり、図13に示した符号化器への入力画像に対応する画像が再構成される。復号された画像の各画素値は、予測画像を生成するために、動き補償器203に備えられた図示せぬフレームメモリに保存される。

【0023】動き補償器203は、図13に示した符号化器の動き補償器103と同一のものであり、逆可変長符号化器209によって得られた動きベクトル $v$ 、およびすでにこの動き補償器203が備える図示せぬフレームメモリに保存されている画像を用いて、現在復号すべき画像の各画素値の予測を行う。

【0024】通常は、上記予測値との差分を符号化する画像間符号化 (INTER coding) を行なう。しかし、現在符号化すべき画素の値と動き補償器103によって算出された予測値との差分が大きい場合、符号化ビット量が多くなることを防ぐために、以下に示す画像内符号化 (INTRA coding) を行なう場合もある。すなわち、ブロック内の各画素値について、差分をとらずにそのままDCT器105に送り、その画素値に対して符号化を行なう。

【0025】以下では、量子化器として用意されている二通りの量子化および逆量子化の処理方法について説明する。

【0026】H. 261/3型の量子化は、例えば図15に示したように、量子化後に0となる範囲が量子化幅に比べて広く設けられているような量子化である (dead zoneの有る量子化)。VM6.0では次式(4)のように行われ、COFはDCT係数、LEVELは量子化後のDCT係数の絶対値、QPは量子化ステップサイズ、QCOFは量子化後のDCT係数を表わす。

【0027】

を表わす。

\* \* 【0033】

For INTRA: COF<sub>2</sub> = |QCOF| × 2 × QP

For INTER: COF<sub>2</sub> = (|QCOF| × 2 + sign(QCOF)) × QP

... (式7)

【0034】但し、H. 261/3型の量子化におけるINTRAの直流成分と、MPEG1/2型の量子化における直流成分については、次式のように量子化される。dcは直流成分のDCT係数、QDCはその量子化後の値を示す。

【0035】QDC = dc // 8 ... (式8)

【0036】この逆量子化は次式のように行われ、dc<sub>2</sub>は逆量子化後の直流成分のDCT係数を表わす。

【0037】dc<sub>2</sub> = QDC × 8 ... (式9)

【0038】DCT係数の量子化を行う際、例えば低周波成分に比べると重要な情報を持たないと見られる高周波成分をより多く無為化するために、量子化行列を設定※

COF[i][j] = (16 × COF[i][j]) // w[i][j]

... (式10)

【0040】量子化行列を用いた場合には、逆量子化後の

係数を例えば次式(11)のように再調整する必要がある。COF<sub>3</sub>[i][j]は再調整後のi行j列番目★

COF<sub>3</sub>[i][j] = COF<sub>2</sub>[i][j] × w[i][j] / 16

... (式11)

【0042】エンコーダ側とデコーダ側とでIDCT器の精度や丸めの方法が異なる場合、符号化を行うにつれて不整合(ミスマッチ)が生じ、次第にその誤差が視覚的に感知されるようになる。この問題を防ぐために、IDCTを行う前にミスマッチ対策が行われ、その対策方☆

if COF<sub>2</sub> is odd: |COF<sub>4</sub>| = |COF<sub>2</sub>|

if COF<sub>2</sub> is even: |COF<sub>4</sub>| = |COF<sub>2</sub>| - 1

COF<sub>4</sub> = sign(COF<sub>2</sub>) × |COF<sub>4</sub>| ... (式12)

【0044】他方はDCT係数の最も高周波成分を操作するミスマッチ対策で、DCT係数の奇数化によるミスマッチ対策よりも効率が良く、画質の劣化も少ない。縦横8要素のDCT係数のブロックについては、DCT係数の7行7列番目の成分、つまり(7, 7)要素を次式◆

sum =  $\sum_{i=0}^{i<8} \sum_{j=0}^{j<8} \text{COF}[i][j]$

if sum is odd:

if sum is even & COF[7][7] is odd:

if sum is even & COF[7][7] is even:

※して各係数の重みを調整することができる。この量子化行列は、例えば図17に示すような縦横8要素の行列であり、図17(A)に示した画像内符号化と図17

(B)に示した画像間符号化とは異なる行列が設定されるのが普通である。量子化行列を用いる場合、VM6.0では次式(10)のようにDCT係数が処理される。COF[i][j]はDCT係数のi行j列番目の成分、w[i][j]は量子化行列のi行j列番目の重み係数、COF<sub>2</sub>[i][j]は量子化行列による重み調整を行った後のi行j列番目のDCT係数を表わす。

【0039】

20★のDCT係数を表わす。

【0041】

☆法として以下に示す二通りの処理方法がよく用いられている。一方はDCT係数を奇数化することによるミスマッチ対策で、例えば次式(12)のように行われる。COF<sub>4</sub>はミスマッチ対策後のDCT係数を表わす。

【0043】

◆のようにtoggleする。sumはDCT係数の和、COF<sub>7</sub>[7][7]はミスマッチ対策後のDCT係数の(7, 7)要素を表わす。

【0045】

【数1】

COF<sub>m</sub>[7][7] = COF[7][7]

COF<sub>m</sub>[7][7] = COF[7][7] - 1

COF<sub>m</sub>[7][7] = COF[7][7] + 1

... (式13)

【0046】なお、逆量子化された係数はIDCTの前でクリッピングされ、VM6.0ではその制限される範囲は[-2048:2047]である。

【0047】次に、従来例におけるシンタックスを説明する。以下MPEG4 VM (Verification Model)を

例にし、説明する。

【0048】図18に符号化ビット列の構成を示す。VS (Video Session Class) は一つ又は複数のVO (Video Object Class) から構成される、符号化ビット列の集合である。VSのシンタックスを図19に示す。

【0049】図20にVO (Video Object Class) のシンタックスを示す。VOは画像全体または画像中の一部の物体の符号化ビット列である。

【0050】VOL (Video Object Layer Class) はスケーラビリティのためのクラスである。VOLのシンタックスを図21、図22に示す。VOLはvideo\_object\_layer\_idに示される番号によって識別される。例えば、video\_object\_layer\_id=0であるVOL0は下位レイヤであり、例えばvideo\_object\_layer\_id=1であるVOL1は上位レイヤである。スケーラブルのレイヤの数は任意で良い。

【0051】また、video\_object\_layer\_quant\_typeは、量子化等を行なう際に用いる処理方法の組合せのモードを示すフラグである。

【0052】図23乃至図26にVOP (Video Object Plane Class) のシンタックスを示す。VOP\_widthおよびVOP\_heightはそのVOPの大きさを示すフラグである。

【0053】図27はIおよびPピクチャ (VOP) におけるマクロブロック (MacroblockLayer) のシンタックスを示している。CODはこれ以後そのマクロブロックのデータが存在するかどうかを示すフラグである。COD=1の場合、これ以後、そのマクロブロックのデータが存在しないことを示す (即ちスキップマクロブロック)。COD=0の場合、さらにフラグが伝送される。MCBPCはそのマクロブロックのタイプを示すフラグで、これに従って所定のフラグおよびデータが伝送される。

【0054】図28はBピクチャ (VOP) におけるマクロブロック (Macroblock Layer) のシンタックスを示している。もし、最も最近に復号されたIまたはP VOPの対応するマクロブロックがスキップマクロブロックであった場合 (COD=1)、B VOPにおけるそのマクロブロックもスキップマクロブロックとなる。MODBはB VOPにおいてマクロブロックのタイプを示すフラグである。MODBの可変長符号を図29に示す。MODB=0の場合、それ以上マクロブロックのデータが存在しないことを示す。MODB=10の場合、CBPBは伝送されず、MBTYPEが伝送される。MODB=11の場合、CBPBおよびMBTYPEが伝送される。

【0055】CBPBは6ビットのフラグでマクロブロック中の各ブロックにDCT係数が存在するかどうかを示すフラグである。CBPBが伝送されない場合、CBPBは0であると解釈され、そのマクロブロックではDCT係数は伝送されない。MBTYPEはBピクチャにおける各マクロブロックの予測モードを示すフラグで、これに従って所定のフラグおよびデータが伝送される。

【0056】量子化等を中心として、従来例における処理の流れを詳しく説明するため、図30、図31に、従

来例における処理の流れと符号化ビット列の生成法および解読法の一例を示す。図30に示したフローチャートは、エンコーダの量子化およびミスマッチ制御部分を示し、図31に示したフローチャートは、デコーダのミスマッチ制御および逆量子化部分を示している。

【0057】図30のフローチャートを参照して、エンコーダの量子化およびミスマッチ制御部分における処理の流れと符号化ビット列の生成法を説明する。まず、ステップS1において、何らかの評価基準を用いて、H. 261/3、MPEG1/2のどちらのモードで量子化等を行なうかを定める。あるいはエンコードする際に、あらかじめどのモードの量子化等を行なうかの指定がなされていても良い。次に、DCT演算を行なって求めた縦横8要素のDCT係数の各ブロックに対して、各モードに応じて以下のように量子化行列の利用方法を設定し、量子化および逆量子化の方法を選択し、IDCTミスマッチの対策方法を選択する。

【0058】H. 261/3モードの場合は、ステップS2において、量子化器121により、縦横8要素のDCT係数の各ブロックに対してH. 261/3型の量子化が行われ、逆量子化器123により、逆量子化が行われる。その後、ステップS3に進み、ミスマッチ制御器141により、DCT係数を奇数化するミスマッチ制御が行われ、ステップS4において、クリッピング器143によりDCT係数のクリッピングが行われる。このモードの場合には、量子化行列は利用しないため、量子化行列は設定されずに量子化および逆量子化が行なわれる。このモードが選ばれた場合には、ステップS5において、Video Object Layer Classのvideo\_object\_layer\_quant\_typeのフラグは"0"に設定され、符号化される。

【0059】一方、MPEG1/2モードの場合は、ステップS6に進み、設定器132により、縦横8要素のDCT係数の各ブロックに対してまずMPEG1/2型の量子化行列が設定される。ここで設定される量子化行列は、画像内符号化と画像間符号化とは異なる。また、あらかじめ用意されている量子化行列以外の行列を新たに設定して用いても良いが、その場合には、Video Object Layer Classのload\_intra\_quant\_mat、あるいはload\_nonintra\_quant\_matのフラグをONにし、新たに設定した量子化行列の64個の要素をintra\_quant\_mat[64]あるいはnonintra\_quant\_mat[64]として符号化する。

【0060】その後、ステップS7において、量子化器122により、ステップS6において設定器132により設定された量子化行列を用いてMPEG1/2型の量子化が行われ、逆量子化器124により、逆量子化が行われる。次に、ステップS8に進み、クリッピング器144により、DCT係数のクリッピングが行われる。ここでは、ミスマッチ制御は行なわれないことに注意されたい。このモードが選ばれた場合には、ステップS9において、Video Object Layer Classのvideo\_object\_layer



er\_quant\_typeのフラグは"1"に設定され、符号化される。

【0061】次に、図31に示したフローチャートを参照して、デコーダのミスマッチ制御および逆量子化部分における処理の流れと符号化ビット列の解読法を説明する。最初に、ステップS11において、図13に示したエンコーダによって符号化された符号化ビット列のうち、Video Object Layer Classのvideo\_object\_layer\_quant\_typeのフラグを解読することによって、エンコーダ側で用いたモードを検出し、エンコーダ側のモードに10 対応するように逆量子化等の処理を行なう。例えば、video\_object\_layer\_quant\_type=="0" ("=="は、左辺と右辺の値が等しいことを意味する) の場合にはH. 261/3モード、そうでない場合にはMPEG1/2モードとみなして処理を行なう。

【0062】H. 261/3モードの場合は、ステップS12に進み、この場合、量子化行列は利用しないため、量子化行列は設定されずに逆量子化が行なわれる。即ち、復号された縦横8要素のDCT係数の各ブロックに対して、まず逆量子化器223により、H. 261/3型の逆量子化が行われる。その後、ステップS13において、ミスマッチ制御部241により、DCT係数を奇数化するミスマッチ制御が行われる。次に、ステップS14に進み、クリッピング器243により、DCT係数のクリッピングが行われる。

【0063】一方、MPEG1/2モードの場合は、ステップS15において、復号された縦横8要素のDCT係数の各ブロックに対して、まず設定器232によりMPEG1/2型の量子化行列が設定される。ここで設定される量子化行列は、画像内符号化と画像間符号化とでは異なる。また、例えばVideo Object Layer Classのload\_intra\_quant\_mat、あるいはload\_nonintra\_quant\_matのフラグがONになっている場合には、あらかじめ用意されている量子化行列以外を設定するために、例えば、intra\_quant\_mat[64]あるいはnonintra\_quant\_mat[64]として符号化された量子化行列の64個の要素を復号して用いる等の処理を行なわなければならない。その後、ステップS16に進み、逆量子化器224により、ステップS15において設定された量子化行列を用いてMPEG1/2型の逆量子化が行われ、ステップS17において、クリッピング器244によりDCT係数のクリッピングが行われる。エンコーダの場合と同様、ミスマッチ制御は行なわれないことに注意されたい。

【0064】以上、MPEG方式について説明したが、この他にも様々な動画画像の高効率符号化方法が標準化されている。例えば、ITU-Tでは主に通信用の符号化方式として、H. 261やH. 263という方式を規定している。このH. 261やH. 263も基本的にはMPEG方式と同様に動き補償予測符号化とDCT変換符号化を組み合わせたものであり、ヘッダ情報などの詳細

は異なるが、符号化装置や復号装置は同様の構成となる。

#### 【0065】

【発明が解決しようとする課題】しかしながら、従来のMPEG1、MPEG2、H. 261、H. 263等の動画画像の高効率符号化方法では、量子化および逆量子化の方法、量子化行列の利用方法、IDCTミスマッチの対策方法等としてそれぞれ異なる方法が指定されており、その他の方法を用いることは許されていなかった。

【0066】MPEG4においては、1997年3月7日の時点で、これらの既存の動画画像の高効率符号化方法との互換性をどのようにするかについては特にまとめられていない。そのため、MPEG4のVM (Verification Model) やWD (WorkingDraft) の規定では、量子化および逆量子化の方法、量子化行列の利用方法、IDCTミスマッチの対策方法等が、暫定的に複数用意されている。符号化および復号する場合には、これらの方法の中で決められた方法の組合せのモードを用いなければならない。規定されている組合せ自体も充分ではなかった。

【0067】ここで、MPEG4のVM6.0 (ISO/IEC JTC1/SC29/WG11 MPEG96/N1582 Sevilla, February 1996) や、WD1.0 (ISO/IEC JTC1/SC29/WG11 MPEG96/N1470Maceio, Nov. 1996) で規定されている符号化および復号方法のうち、量子化およびミスマッチ制御の部分を簡単に説明する。H. 261、あるいはH. 263 (以下、H. 261/3と示す) のモードで符号化および復号する場合には、量子化および逆量子化の方法、量子化行列の利用方法、IDCTミスマッチの対策方法が全て指定される。MPEG1あるいはMPEG2 (以下、MPEG1/2と示す) のモードで符号化および復号する場合には、量子化および逆量子化の方法、量子化行列の利用方法は指定されるが、IDCTミスマッチの対策方法は指定されない。

【0068】このように、従来の動画画像の高効率符号化方法では、既存の動画画像の高効率符号化方法との互換性が充分に取られていない課題があった。

【0069】また、量子化および逆量子化の方法、量子化行列の利用方法、IDCTミスマッチの対策方法等で、最適な方法の組合せを指定して、より高効率の符号化および復号を行なうこともできない課題があった。

【0070】本発明はこのような状況に鑑みてなされたものであり、既存の動画画像の高効率符号化方法との互換性を取ることができるようにするものである。

#### 【0071】

【課題を解決するための手段】請求項1に記載の符号化装置は、データを符号化する符号化装置であって、データを複数の方法で量子化する量子化手段と、量子化手段の出力を複数の方法で逆量子化する逆量子化手段と、量子化手段および逆量子化手段で用いる量子化行列を設定する設定手段と、逆量子化手段の出力に対して複数の方

法でIDCTミスマッチ対策を施すミスマッチ対策手段と、量子化手段がデータを量子化する方法の任意のものと、逆量子化手段が量子化手段の出力を逆量子化する方法の任意のものと、ミスマッチ対策手段が逆量子化手段の出力に対してIDCTミスマッチ対策を施す方法の任意のものと、量子化手段および逆量子化手段が量子化行列を使用する任意の方法とを組み合わせることでデータを符号化する方法を制御する制御手段とを備えることを特徴とする。

【0072】請求項2に記載の符号化方法は、データを符号化する符号化方法であって、データを複数の方法で量子化するステップと、量子化されたデータを複数の方法で逆量子化するステップと、量子化および逆量子化を行うときに用いる量子化行列を設定するステップと、逆量子化されたデータに対して複数の方法でIDCTミスマッチ対策を施すステップと、データを量子化する方法の任意のものと、量子化されたデータを逆量子化する方法の任意のものと、逆量子化されたデータに対してIDCTミスマッチ対策を施す方法の任意のものと、データを量子化するとき、および量子化されたデータを逆量子化するとき量子化行列を使用する任意の方法とを組み合わせることでデータを符号化する方法を制御するステップとを備えることを特徴とする。

【0073】請求項3に記載の復号装置は、データを復号する復号装置であって、データを複数の方法で逆量子化する逆量子化手段と、逆量子化手段で用いる量子化行列を設定する設定手段と、逆量子化手段の出力に対して複数の方法でIDCTミスマッチ対策を施すミスマッチ対策手段と、逆量子化手段がデータを逆量子化する方法の任意のものと、ミスマッチ対策手段が逆量子化手段の出力に対してIDCTミスマッチ対策を施す方法の任意のものと、逆量子化手段が量子化行列を使用する任意の方法とを組み合わせることでデータを復号する方法を制御する制御手段とを備えることを特徴とする。

【0074】請求項4に記載の復号方法は、データを復号する復号方法であって、データを複数の方法で逆量子化するステップと、データを逆量子化するとき用いる量子化行列を設定するステップと、逆量子化されたデータに対して複数の方法でIDCTミスマッチ対策を施すステップと、データを逆量子化する方法の任意のものと、逆量子化されたデータに対してIDCTミスマッチ対策を施す方法の任意のものと、データを逆量子化するとき量子化行列を使用する任意の方法とを組み合わせることでデータを復号する方法を制御するステップとを備えることを特徴とする。

【0075】請求項5に記載の復号装置は、データを復号する復号装置であって、データを複数の方法で逆量子化する逆量子化手段と、逆量子化手段で用いる量子化行列を設定する設定手段と、逆量子化手段の出力に対して所定の方法でIDCTミスマッチ対策を施すミスマッチ

対策手段と、逆量子化手段がデータを逆量子化する方法の任意のものと、逆量子化手段が量子化行列を使用する任意の方法とを組み合わせることでデータを復号する方法を制御する制御手段とを備えることを特徴とする。

【0076】請求項6に記載の復号方法は、データを復号する復号方法であって、データを複数の方法で逆量子化するステップと、データを逆量子化するとき用いる量子化行列を設定するステップと、逆量子化されたデータに対して所定の方法でIDCTミスマッチ対策を施すステップと、データを逆量子化する方法の任意のものと、データを逆量子化するとき量子化行列を使用する任意の方法とを組み合わせることでデータを復号する方法を制御するステップとを備えることを特徴とする。

【0077】請求項1に記載の符号化装置においては、量子化手段が、データを複数の方法で量子化し、逆量子化手段が、量子化手段の出力を複数の方法で逆量子化し、設定手段が、量子化手段および逆量子化手段で用いる量子化行列を設定し、ミスマッチ対策手段が、逆量子化手段の出力に対して複数の方法でIDCTミスマッチ対策を施し、制御手段が、量子化手段がデータを量子化する方法の任意のものと、逆量子化手段が量子化手段の出力を逆量子化する方法の任意のものと、ミスマッチ対策手段が逆量子化手段の出力に対してIDCTミスマッチ対策を施す方法の任意のものと、量子化手段および逆量子化手段が量子化行列を使用する任意の方法とを組み合わせることでデータを符号化する方法を制御する。

【0078】請求項2に記載の符号化方法においては、データを複数の方法で量子化し、量子化されたデータを複数の方法で逆量子化し、量子化および逆量子化を行うときに用いる量子化行列を設定し、逆量子化されたデータに対して複数の方法でIDCTミスマッチ対策を施すとき、データを量子化する方法の任意のものと、量子化されたデータを逆量子化する方法の任意のものと、逆量子化されたデータに対してIDCTミスマッチ対策を施す方法の任意のものと、データを量子化するとき、および量子化されたデータを逆量子化するとき量子化行列を使用する任意の方法とを組み合わせることでデータを符号化する方法を制御する。

【0079】請求項3に記載の復号装置においては、逆量子化手段が、データを複数の方法で逆量子化し、設定手段が、逆量子化手段で用いる量子化行列を設定し、ミスマッチ対策手段が、逆量子化手段の出力に対して複数の方法でIDCTミスマッチ対策を施し、制御手段が、逆量子化手段がデータを逆量子化する方法の任意のものと、ミスマッチ対策手段が逆量子化手段の出力に対してIDCTミスマッチ対策を施す方法の任意のものと、逆量子化手段が量子化行列を使用する任意の方法とを組み合わせることでデータを復号する方法を制御する。

【0080】請求項4に記載の復号方法においては、データを複数の方法で逆量子化し、データを逆量子化する

とき用いる量子化行列を設定し、逆量子化されたデータに対して複数の方法でIDCTミスマッチ対策を施すとき、データを逆量子化する方法の任意のものと、逆量子化されたデータに対してIDCTミスマッチ対策を施す方法の任意のものと、データを逆量子化するとき量子化行列を使用する任意の方法を組み合わせるデータを復号する方法を制御する。

【0081】請求項5に記載の復号装置においては、逆量子化手段が、データを複数の方法で逆量子化し、設定手段が、逆量子化手段で用いる量子化行列を設定し、ミスマッチ対策手段が、逆量子化手段の出力に対して所定の方法でIDCTミスマッチ対策を施し、制御手段が、逆量子化手段がデータを逆量子化する方法の任意のものと、逆量子化手段が量子化行列を使用する任意の方法を組み合わせるデータを復号する方法を制御する制御手段とを備える。

【0082】請求項6に記載の復号方法においては、データを複数の方法で逆量子化し、データを逆量子化するとき用いる量子化行列を設定し、逆量子化されたデータに対して所定の方法でIDCTミスマッチ対策を施すとき、データを逆量子化する方法の任意のものと、データを逆量子化するとき量子化行列を使用する任意の方法を組み合わせるデータを復号する方法を制御する。

【0083】

【発明の実施の形態】以下に、本発明の実施の形態を説明するが、その前に、特許請求の範囲に記載の発明の各手段と以下の実施の形態との対応関係を明らかにするために、各手段の後の括弧内に、対応する実施の形態（但し、一例）を付加して、本発明の特徴を記述すると、次のようになる。

【0084】即ち、請求項1に記載の符号化装置は、データを符号化する符号化装置であって、データを複数の方法で量子化する量子化手段（例えば、図1の量子化器121、122）と、量子化手段の出力を複数の方法で逆量子化する逆量子化手段（例えば、図1の逆量子化器123、124、125）と、量子化手段および逆量子化手段で用いる量子化行列を設定する設定手段（例えば、図1の設定器132）と、逆量子化手段の出力に対して複数の方法でIDCTミスマッチ対策を施すミスマッチ対策手段（例えば、図1のミスマッチ制御器141、151、154）と、量子化手段がデータを量子化する方法の任意のものと、逆量子化手段が量子化手段の出力を逆量子化する方法の任意のものと、ミスマッチ対策手段が逆量子化手段の出力に対してIDCTミスマッチ対策を施す方法の任意のものと、量子化手段および逆量子化手段が量子化行列を使用する任意の方法とを組み合わせるデータを符号化する方法を制御する制御手段（例えば、図1のスイッチSW1、SW2、SW21、SW22）とを備えることを特徴とする。

【0085】請求項3に記載の復号装置は、データを復

号する復号装置であって、データを複数の方法で逆量子化する逆量子化手段（例えば、図2の逆量子化器223、224、225）と、逆量子化手段で用いる量子化行列を設定する設定手段（例えば、図2の設定器232、233）と、逆量子化手段の出力に対して複数の方法でIDCTミスマッチ対策を施すミスマッチ対策手段（例えば、図2のミスマッチ制御器241、251、253）と、逆量子化手段がデータを逆量子化する方法の任意のものと、ミスマッチ対策手段が逆量子化手段の出力に対してIDCTミスマッチ対策を施す方法の任意のものと、逆量子化手段が量子化行列を使用する任意の方法を組み合わせるデータを復号する方法を制御する制御手段（例えば、図2のスイッチSW31、SW32）とを備えることを特徴とする。

【0086】請求項5に記載の復号装置は、データを復号する復号装置であって、データを複数の方法で逆量子化する逆量子化手段（例えば、図11の逆量子化器223、224）と、逆量子化手段で用いる量子化行列を設定する設定手段（例えば、図11の設定器232）と、逆量子化手段の出力に対して所定の方法でIDCTミスマッチ対策を施すミスマッチ対策手段（例えば、図11のミスマッチ制御器241）と、逆量子化手段がデータを逆量子化する方法の任意のものと、逆量子化手段が量子化行列を使用する任意の方法を組み合わせるデータを復号する方法を制御する制御手段（例えば、図11のスイッチSW11、SW51、SW52、SW53）とを備えることを特徴とする。

【0087】なお、勿論この記載は、各手段を上記したものに限定することを意味するものではない。

【0088】図1に本発明の符号化装置を応用した符号化器（エンコーダ）の一実施の形態の構成例を示す。また、図2に、本発明の復号装置を応用した復号器（デコーダ）の一実施の形態の構成例を示す。図1はエンコーダ、図2はデコーダを示している。図1に示した符号化器の基本的な構成は図13に示した従来例の場合とほぼ同様である。また、図2に示した復号器の基本的な構成は図14に示した従来例の場合とほぼ同様である。

【0089】従来例と異なる点は、エンコーダについては図13の符号化器において、スイッチSW3、SW4で分岐される処理のモードが、図1の符号化器においては、スイッチSW21、SW22においてそれぞれ三通り存在する点と、MPEG1あるいはMPEG2モードの場合、逆量子化を行なった後にミスマッチ制御を行なう部分が存在する点である。デコーダについても、図2の復号器においては、スイッチSW31、SW32で分岐される処理のモードが三通り存在する点と、MPEG1あるいはMPEG2モードの場合、逆量子化を行なった後にミスマッチ制御を行なう部分が存在する点である。

【0090】図1の符号化器におけるスイッチSW1、

SW2, SW21, SW22、図2の復号器におけるスイッチSW31, SW32は、図13, 図14に示した従来例の場合と同様に、どのモードの量子化等を行なうかを示すフラグQ\_mode2と連動しており、図の例ではH. 261/3モードの場合を示した。但し、符号化器におけるスイッチSW1, SW2について、MPEG1あるいはMPEG2モードの場合の処理は共通である。フラグQ\_mode2は、量子化等の処理方法の組み合わせを示しており、符号化器においては、あらかじめ何らかの評価基準を用い、H. 261/3, MPEG1, MPEG2のうち、どのモードで量子化等を行なうかを決めておく。あるいはエンコードする際に、あらかじめどのモードの量子化等を行なうかの指定がなされていても良い。図2の復号器においては、逆可変長符号化器209によって復号される値による。

【0091】逆量子化後に行なうミスマッチ制御に関連して、DCT係数のクリッピング器(図1の符号化器ではクリッピング器143, 152, 153、図2の復号器ではクリッピング器243, 244, 252)の位置がMPEG1モードとMPEG2モードとで前後していることに注意されたい。MPEG1モード(およびH. 261/3モード)では、DCT係数の奇数化によるミスマッチ制御器(図1の符号化器ではミスマッチ制御器141, 151、図2の復号器ではミスマッチ制御器241, 251)の後にDCT係数のクリッピング器が置かれる。

【0092】これに対して、MPEG2モードでは、DCT係数のクリッピング器の後に、DCT係数の(7, 7)係数をtoggleするミスマッチ制御器(図1の符号化器ではミスマッチ制御器154、図2の復号器ではミスマッチ制御器253)が置かれる。

【0093】以下、図1、図2の実施の形態における、より詳しい処理の流れについて説明する。

【0094】図3および図4に、図1および図2に示した符号化器および復号器における処理の流れと符号化ビット列の生成法および解読法の一例を示す。図3はエンコーダの量子化およびミスマッチ制御部分を示し、図4はデコーダのミスマッチ制御および逆量子化部分を示した。

【0095】上記実施の形態における符号化シンタックスの基本的構成は、図30、図31に示した従来例の場合と基本的に同一であるが、量子化および逆量子化の方法等の組合せとして指定可能なモードとして、H. 261/3モード以外にMPEG1, MPEG2モードの三つが用意されている点が大きく異なる。またMPEG1, MPEG2モードでは、IDCTミスマッチの対策方法が指定されている点も異なる。

【0096】上記実施の形態におけるVideo Object Layer Classのシンタックスの一部を図5に示した。従来例とは、video\_object\_layer\_quant\_typeのフラグが2bit

である点異なる。

【0097】次に、図3のフローチャートを参照して、図1に示したエンコーダの量子化およびミスマッチ制御部分における処理の流れと符号化ビット列の生成法を説明する。まず、ステップS21において、何らかの評価基準を用いて、H. 261/3, MPEG1, MPEG2のうちどのモードで量子化等を行なうかを定める。あるいはエンコードする際に、あらかじめどのモードの量子化等を行なうかの指定がなされていても良い。DCT演算を行なって求めた縦横8要素のDCT係数の各ブロックに対して、各モードに応じて以下のように量子化行列の利用方法を設定し、量子化および逆量子化の方法を選択し、IDCTミスマッチの対策方法を選択する。

【0098】H. 261/3モードの場合は、ステップS22において、量子化器121により、縦横8要素のDCT係数の各ブロックに対して、H. 261/3型の量子化が行われ、次に、逆量子化器123により逆量子化が行われる。その後、ステップS23に進み、ミスマッチ制御器141により、DCT係数を奇数化するミスマッチ制御が行われ、次に、ステップS24において、クリッピング器143によりDCT係数のクリッピングが行われる。

【0099】このモードの場合には、量子化行列は利用しないため、量子化行列は設定されずに量子化および逆量子化が行なわれる。なお、このモードが選ばれた場合には、ステップS25において、例えば、Video Object Layer Classのvideo\_object\_layer\_quant\_typeのフラグは"01"に設定され、符号化される。この例では、このフラグの名前は従来例の場合と同名になっているが、以下にも説明するように、その役割は従来例とは異なる。

【0100】一方、MPEG1モードの場合は、ステップS26において、設定器132により、縦横8要素のDCT係数の各ブロックに対してまず、MPEG1/2型の量子化行列が設定される。ここで設定される量子化行列は、画像内符号化と画像間符号化とは異なる。また、あらかじめ用意されている量子化行列以外のものを新たに設定して用いても良いが、その場合には、例えばVideo Object Layer Classのload\_intra\_quant\_mat、あるいはload\_nonintra\_quant\_matのフラグをONにし、新たに設定した量子化行列の64個の要素をintra\_quant\_mat[64]あるいはnonintra\_quant\_mat[64]として符号化する等の処理を行なわなければならない。

【0101】その後、ステップS27において、量子化器122により、設定器132が設定した量子化行列を用いてMPEG1/2型の量子化が行われ、次に、逆量子化器124により、逆量子化が行われる。次に、ステップS28に進み、ミスマッチ制御器151により、DCT係数を奇数化するミスマッチ制御が行われ、ステップS29において、クリッピング器152により、DCT係数のクリッピングが行われる。なお、このモードが

選ばれた場合には、ステップS30において、例えば、Video Object Layer Classのvideo\_object\_layer\_quant\_typeのフラグは"10"に設定され、符号化される。

【0102】さらに、MPEG2モードの場合は、ステップS31において、設定器132により、縦横8要素のDCT係数の各ブロックに対して、MPEG1モードの場合と同様に、まずMPEG1/2型の量子化行列が設定される。ここで、新たに量子化行列を設定して用いる場合にも、MPEG1モードの場合と同様に量子化行列の64個の要素を符号化する等の処理を行なわなければならない。

【0103】その後、ステップS32に進み、設定器132が設定した量子化行列を用いてMPEG1/2型の量子化が行われ、逆量子化器125により、逆量子化が行われる。次に、ステップS33において、クリッピング器153により、DCT係数のクリッピングが行われてから、ステップS34に進み、ミスマッチ制御器154により、DCT係数の(7, 7)要素をtoggleすることによるミスマッチ制御が行われる。なお、このモードが選ばれた場合には、ステップS35において、例えば、Video Object Layer Classのvideo\_object\_layer\_quant\_typeのフラグは"11"に設定され、符号化される。ステップS25またはステップS30またはステップS35の処理が終了すると、すべての処理を終了する。

【0104】次に、図4のフローチャートを参照して、図2に示したデコーダのミスマッチ制御および逆量子化部分における処理の流れと符号化ビット列の読法について説明する。ステップS31においては、図1に示したエンコーダによって符号化された符号化ビット列のうち、例えば、Video Object Layer Classのvideo\_object\_layer\_quant\_typeのフラグを解読することによって、エンコーダ側で用いたモードに対応するように逆量子化等の処理を行なう。例えば、video\_object\_layer\_quant\_type=="01"の場合にはH. 261/3モード、video\_object\_layer\_quant\_type=="10"の場合にはMPEG1モード、video\_object\_layer\_quant\_type=="11"の場合にはMPEG2モードとみなして処理を行なう。

【0105】H. 261/3モードの場合は、量子化行列は利用しないため、ステップS32に進み、量子化行列は設定されずに逆量子化が行なわれる。即ち、復号された縦横8要素のDCT係数の各ブロックに対して、まず、逆量子化器223により、H. 261/3型の逆量子化が行われる。その後、ステップS33において、ミスマッチ制御器141により、DCT係数を奇数化するミスマッチ制御が行われ、ステップS34に進み、クリッピング器243により、DCT係数のクリッピングが行われる。

【0106】また、MPEG1モードの場合は、復号された縦横8要素のDCT係数の各ブロックに対して、まず、ステップS35において、設定器232により、M

PPEG1/2型の量子化行列が設定される。ここで設定される量子化行列は、画像内符号化と画像間符号化とは異なる。また、例えば、Video Object Layer Classのload\_intra\_quant\_mat、あるいはload\_nonintra\_quant\_matのフラグがONになっている場合には、あらかじめ用意されている量子化行列以外のものを設定するために、例えば、intra\_quant\_mat[64]あるいはnonintra\_quant\_mat[64]として符号化された量子化行列の64個の要素を復号して用いる等の処理を行なわなければならない。次に、ステップS36に進み、逆量子化器224により、設定器232が設定した量子化行列を用いてMPEG1/2型の逆量子化が行われる。その後、ステップS37において、ミスマッチ制御器251により、DCT係数を奇数化するミスマッチ制御が行われ、次に、ステップS38において、クリッピング器244により、DCT係数のクリッピングが行われる。

【0107】さらに、MPEG2モードの場合は、復号された縦横8要素のDCT係数の各ブロックに対して、まず、ステップS39において、設定器233により、MPEG1/2型の量子化行列が設定される。ここで設定される量子化行列は、MPEG1モードの場合と同様であり、新たに量子化行列が設定されている場合には、符号化された量子化行列を復号して用いる等の処理を行なわなければならない。次に、ステップS40に進み、逆量子化器225により、設定器233が設定した量子化行列を用いてMPEG1/2型の逆量子化が行われる。その後、ステップS41において、クリッピング器252により、DCT係数のクリッピングが行われ、ステップS42において、ミスマッチ制御器253により、DCT係数の(7, 7)要素をtoggleすることによるミスマッチ制御が行われる。

【0108】上記処理以外でも、H. 261/3、MPEG1、MPEG2の各モードにおいて個別に処理すべき他の処理を、例えば、Video Object Layer Classのvideo\_object\_layer\_quant\_typeのフラグと連動させることにしても良い。あるいは、新たなモードを設け、そのモードと対応するような新規あるいは従来の量子化および逆量子化等の各処理方法を連動させることにしても良い。あるいは、逆に用意するモードの数自体を減らしたり、フラグと連動させる処理の数を減らすなど、処理を簡潔にしても良い。あるいは、可能である限りこれらの処理の順番を入れ換えても良い。どのモードの処理方法を行なうかを示す例えば、video\_object\_layer\_quant\_typeのフラグで用いるシンボルは、上記の例とは異なったシンボルを用いても良い。

【0109】また上記の例では、各モードを示すフラグとして、例えばVideo Object LayerClassのvideo\_object\_layer\_quant\_typeのフラグとして説明を行なったが、同様の役割を担う別名のフラグであっても良い。あるいは、例えば、Video ObjectLayer Classにおいてこのフ

ラグを設定する代わりに、Video Object Plane Class又はMacroblock Layer等の他のLayerで設定することによって、設定したフラグに連動させる処理方法をオブジェクト単位からフレーム又はマクロブロック単位等の別の単位で切り替えることにしても良い。

【0110】図6、図7に、本発明の符号化装置および復号装置を適用したエンコーダ（符号化器）およびデコーダ（復号器）の一実施の形態の構成例を示す。図5はエンコーダ、図6はデコーダを示している。図6、図7に示したエンコーダおよびデコーダの基本的な構成は、

図1、図2に示した実施の形態の場合とほぼ同様である。

【0111】図1および図2の実施の形態と異なる点は、エンコーダについては図6の符号化器において、スイッチSW1、SW41、SW42、SW2、SW3、SW43、SW44、SW45とフラグQ\_matrix、Q\_type、mismatchによって、量子化行列の利用方法、量子化および逆量子化の方法、IDCTミスマッチの対策方法を個別に設定することが可能となっている点である。デコーダについても図7の復号器において、スイッチSW11、SW51、SW52、SW53、SW54、SW55とフラグQ\_matrix、Q\_type、mismatchによって、量子化行列の利用方法、逆量子化の方法、IDCTミスマッチの対策方法を個別に設定することが可能となっている点である。

【0112】本実施の形態では、図1、図2に示した実施の形態におけるフラグQ\_mode2に代って、Q\_matrix、Q\_type、mismatchと名付けられた3つのフラグが用意されている。フラグQ\_matrixは、図6の符号化器においてはスイッチSW1、SW41、図7の復号器においてはスイッチSW11、SW51と連動しており、量子化行列の利用方法に係わる。フラグQ\_typeは、図6の符号化器においてはスイッチSW42、SW2、SW3、SW43、図7の復号器においてはスイッチSW52、SW53と連動しており、量子化および逆量子化の方法に係わる。

【0113】フラグmismatchは、図6の符号化器においてはスイッチSW44、SW45、図7の復号器においてはスイッチSW54、SW55と連動しており、IDCTミスマッチの対策方法に係わる。図の例では、Q\_matrixについてはMPEG1/2型の量子化行列を用いない場合、Q\_typeについてはH.261/3型の量子化を行なう場合、mismatchについてはDCT係数の奇数化によるミスマッチ制御を行なう場合を示した。

【0114】3つのフラグQ\_matrix、Q\_type、mismatchは、処理方法の組み合わせを示しており、図6の符号化器においてはあらかじめ何らかの評価基準を用い、それぞれの処理方法を用いるかを決めておく。あるいはエンコードする際に、あらかじめどの処理方法を用いるかの指定がなされていても良い。図7の復号器において

は、逆可変長符号化器209によって復号される値による。

【0115】フラグmismatchによって切り替わるミスマッチ制御に関連して、DCT係数のクリッピング器（図6の符号化器ではクリッピング器143、153、図7の復号器ではクリッピング器243、252）の位置が、図1、図2に示した実施の形態の場合と同様に前後していることに注意されたい。DCT係数の奇数化によるミスマッチ制御を行なう場合には、DCT係数の奇数化によるミスマッチ制御器（図6の符号化器ではミスマッチ制御器141、図7の復号器ではミスマッチ制御器241）の後にDCT係数のクリッピング器が置かれる。

【0116】これに対してDCT係数の（7，7）係数をtoggleすることによるミスマッチ制御を行なう場合には、DCT係数のクリッピング器の後に、DCT係数の（7，7）係数をtoggleするミスマッチ制御器（図6の符号化器ではミスマッチ制御器154、図7の復号器ではミスマッチ制御器253）が置かれる。

【0117】フラグQ\_matrixによって切り替わる量子化行列の利用方法に関連して、仮の量子化行列の設定器（図6の符号化器では設定器131、図7の復号器では設定器261）について説明する。フラグQ\_matrixによってMPEG1/2型の量子化行列を用いないことが示された場合、本来量子化行列は設定されず、この仮の量子化行列の設定器の存在は、通常、符号化器、復号器で共に無視される。しかし、図6、図7の実施の形態においては、同じ量子化および逆量子化の処理方法に対して、量子化行列が設定されている場合と設定されていない場合のどちらかが起こりうるため、そのどちらの場合にも対処できるように、量子化および逆量子化の処理を複数用意しておくべき場合がある。

【0118】しかしながら、量子化行列の設定の有無に係わらず、量子化および逆量子化の処理を一通りで行なえるように、演算あるいは回路を構成することも可能である。それは、量子化行列を設定する場合に合わせるように、量子化行列を設定しない場合には、全ての要素の値が（0以外でかつ正の値で）等しいなどの仮の量子化行列を設定しておき、量子化および逆量子化の処理を行なう際にはこの仮の量子化行列を用いることである。このように、仮の量子化行列を設定する場合、仮の量子化行列の設定器は、図6の符号化器では設定器131、図7の復号器では設定器261の位置に置かれ、存在することになる。

【0119】図6および図7に示した実施の形態における、より詳しい処理の流れについて、以下に説明する。

【0120】図8および図9のフローチャートは、図6および図7の実施の形態における処理の流れと符号化ビット列の生成法および解読法の一例を示している。図8はエンコーダの量子化およびミスマッチ制御部分を示

し、図9はデコーダのミスマッチ制御および逆量子化部分を示している。

【0121】図6および図7の実施の形態における符号化シンタックスの基本的構成は、図3、図4に示した実施の形態の場合と基本的に同様であるが、量子化行列の利用方法、量子化および逆量子化の方法、およびIDCTミスマッチの対策方法等を連動させて設定するのではなく、個別に設定することが可能となっている点が大きく異なる。

【0122】図6および図7の実施の形態におけるVideo Object Layer Classのシンタックスの一部を図10に示した。従来例とは、video\_object\_layer\_quant\_typeのフラグの代わりに、video\_object\_layer\_quant\_matrix, video\_object\_layer\_quant\_type, video\_object\_layer\_mismatchの3つのフラグが設けられている点が異なる。

【0123】図8のフローチャートを参照して、エンコーダの量子化およびミスマッチ制御部分における処理の流れと符号化ビット列の生成法について説明する。ステップS41において、それぞれ複数の処理方法が用意されている、量子化行列の利用方法、量子化および逆量子化の方法、およびIDCTミスマッチの対策方法等について、どの処理方法を用いるかを、何らかの評価基準を用いて決める。あるいはエンコードする際に、あらかじめどの処理方法の量子化等を行なうかの指定がなされていても良い。DCT演算を行なって求めた縦横8要素のDCT係数の各ブロックに対して、以下のように量子化行列の利用方法を設定し、量子化および逆量子化の方法を選択し、IDCTミスマッチの対策方法を選択する。

【0124】まず、決められた量子化行列の利用方法に従って、量子化行列を設定する。図8の例では、MPEG1/2型の量子化行列を用いない場合には、量子化行列を設定しない。あるいは、ステップS42において、設定器131により、量子化行列を設定した場合と量子化の演算を共通に行なえるように、全ての要素の値が(0以外でかつ正の値で)等しいなどの仮の量子化行列が設定される。一方、MPEG1/2型の量子化行列を用いる場合には、ステップS44において、設定器132により、MPEG1/2型の量子化行列が設定される。ここで設定される量子化行列は、図1および図2に示した実施の形態の場合と同様に、画像内符号化と画像間符号化とは異なる。

【0125】また、あらかじめ用意されている量子化行列以外のものを新たに設定して用いても良いが、その場合には、例えばVideo Object Layer Classのload\_intra\_quant\_mat、あるいはload\_nonintra\_quant\_matのフラグをONにし、新たに設定した量子化行列の64個の要素をintra\_quant\_mat[64]あるいはnonintra\_quant\_mat[64]として符号化する等の処理を行なわなければならない。量子化行列の利用方法を示す例えばVideo Object L

ayer Classのvideo\_object\_layer\_quant\_matrixのフラグは、MPEG1/2型の量子化行列を用いない場合にはON、MPEG1/2型の量子化行列を用いる場合にはOFFに設定され、符号化される。

【0126】次に、ステップS46において、H.261/3型の量子化を行うか否かが決定され、決められた量子化の方法に従って、量子化および逆量子化が行なわれる。図8の例では、H.261/3型の量子化を行なう場合には、ステップS47において、量子化器121により、H.261/3型の量子化が行われ、逆量子化器123により、逆量子化が行われる。

【0127】H.261/3型の量子化を行わない場合には、用いる量子化の方法はMPEG1/2型であるとし、ステップS49において、量子化器122により、量子化が行われ、次に、逆量子化器124により逆量子化が行われる。量子化および逆量子化の処理を行なう際には、量子化行列が設定されている場合にはそれを用い、量子化行列が設定されていない場合にはそれを用いない。あるいは、仮の量子化行列が設定されている場合には、それを用いる。

【0128】量子化および逆量子化の方法を示す例えばVideo Object Layer Classのvideo\_object\_layer\_quant\_typeのフラグは、H.261/3型の量子化を行なう場合にはON、H.261/3型の量子化を行わない場合にはOFFに設定され、符号化される。この例では、このフラグの名前は従来例および図1、図2に示した実施の形態の場合(図3、図4のフローチャートに示したフラグ)と同名になっているが、その役割は従来例、図1、図2の実施の形態のどちらとも異なる。

【0129】その後、ステップS51において、DCT係数の奇数化によるミスマッチ制御を行うか否かが決定され、決められたIDCTミスマッチの対策方法に従って、ミスマッチ制御(およびDCT係数のクリッピング)が行われる。図8の例では、DCT係数の奇数化によるミスマッチ制御を行なう場合には、ステップS52において、ミスマッチ制御器141により、DCT係数の奇数化によるミスマッチ制御が行われた後で、ステップS53に進み、クリッピング器143により、DCT係数のクリッピングが行われる。DCT係数の奇数化によるミスマッチ制御を行わない場合には、用いるミスマッチの対策方法はDCT係数の(7,7)係数をtoggleすることによるミスマッチ制御であるとみなし、ステップS55に進み、クリッピング器153により、DCT係数のクリッピングが行われた後で、ステップS56において、ミスマッチ制御器154により、このミスマッチ制御が行われる。

【0130】IDCTミスマッチの対策方法を示す例えばVideo Object Layer Classのvideo\_object\_layer\_mismatchのフラグは、DCT係数の奇数化によるミスマッチ制御を行なう場合にはONに設定され(ステップS5

4)、DCT係数の奇数化によるミスマッチ制御を行わない場合にはOFFに設定され(ステップS57)、符号化される。ステップS54またはステップS57の処理が終了すると、すべての処理が終了する。

【0131】次に、図9のフローチャートを参照して、デコーダのミスマッチ制御および逆量子化部分における処理の流れと符号化ビット列の解読法について説明する。

【0132】復号された縦横8要素のDCT係数の各ブロックに対して、まずエンコーダ側で用いられた量子化10 行列の利用方法に従って、量子化行列を設定する。ステップS61においては、エンコーダによって符号化された符号化ビット列のうち、例えばVideo Object Layer Classのvideo\_object\_layer\_quant\_matrixのフラグが解読される。そして、そのフラグがONである場合には、量子化行列を設定しない。あるいは、ステップS62において、設定器261により、量子化行列を設定した場合と量子化の演算を共通に行なえるように、全ての要素の値が(0以外でかつ正の値で)等しいなどの仮の量子化20 行列が設定される。そのフラグがOFFである場合には、ステップS63において、設定器232により、MPEG1/2型の量子化行列が設定される。ここで設定される量子化行列は、画像内符号化と画像間符号化とは異なる。

【0133】また、例えばVideo Object Layer Classのload\_intra\_quant\_mat、あるいはload\_nonintra\_quant\_matのフラグがONになっている場合には、あらかじめ用意されている量子化行列以外のものを設定するために、例えばintra\_quant\_mat[64]あるいはnonintra\_quant\_mat[64]として符号化された量子化行列の64個の要素を30 復号して用いる等の処理を行わなければならない。

【0134】次に、エンコーダ側で用いられた量子化の方法に従って、逆量子化を行なう。ステップS64においては、エンコーダによって符号化された符号化ビット列のうち、例えばVideo Object Layer Classのvideo\_object\_layer\_quant\_typeのフラグが解読される。そのフラグがONである場合には、ステップS65に進み、逆量子化器223により、H.261/3型の逆量子化が行われる。一方、そのフラグがOFFである場合には、ステップS66に進み、逆量子化器224によりMPEG140 /2型の逆量子化が行われる。逆量子化を行なう際には、量子化行列が設定されている場合にはそれを用い、量子化行列が設定されていない場合にはそれを用いない。あるいは、仮の量子化行列が設定されている場合には、それを用いる。

【0135】その後、エンコーダ側で用いられたIDCTミスマッチの対策方法に従って、ミスマッチ制御(およびDCT係数のクリッピング)を行なう。ステップS67において、エンコーダによって符号化された符号化ビット列のうち、例えばVideo Object Layer Classのvi50

deo\_object\_layer\_mismatchのフラグが解読される。そのフラグがONである場合には、ステップS68に進み、ミスマッチ制御器241により、DCT係数の奇数化によるミスマッチ制御が行われた後で、ステップS69において、クリッピング器243により、DCT係数のクリッピングが行われる。一方、そのフラグがOFFである場合には、ステップS70に進み、クリッピング器252により、DCT係数のクリッピングが行われた後で、ステップS71において、ミスマッチ制御器253により、DCT係数の(7,7)係数をtoggleすることによるミスマッチ制御が行われる。

【0136】上記の例では、量子化行列の利用方法、量子化および逆量子化の方法、およびIDCTミスマッチの対策方法等について、どれも丁度二通りの処理方法だけある場合を示した。しかし、何らかの方法について三通り以上の処理方法が考えられる場合についても、フラグのON,OFFの二通りのシンボルを三種類以上の表現が可能なシンボルに変え、それらのシンボルに応じて処理方法を切り替えるようにしても良い。

【0137】また、上記処理以外についても、いくつかの処理方法が用意又は想定される何らかの処理のために、例えばVideo Object Layer Classにおいて新たなフラグを設けておき、そのフラグとその処理とを連動させることにしても良い。あるいは、逆に用意又は想定される処理の数自体や、処理方法の数を減らす等の簡潔化を行なうことによって、フラグの数を減らす等のことを行なっても良い。あるいは、可能である限りこれらの処理の順番を入れ替えても良い。

【0138】また上記の例では、各方法を示すフラグとして例えばVideo Object Layer Classのvideo\_object\_layer\_quant\_type等のフラグを用いて説明を行なったが、同様の役割を担う別名のフラグであっても良い。あるいは、例えば、Video Object Layer Classにおいてこのフラグを設定する代わりに、Video Object Plane Class又はMacroblock Layer等の他のLayerで設定することによって、設定したフラグに連動させる処理方法をオブジェクト単位からフレーム又はマクロブロック単位等の別の単位で切り替えることにしても良い。

【0139】図11は、本発明の復号装置を応用した復号器(デコーダ)のさらに他の実施の形態の構成例を示すブロック図である。この実施の形態(デコーダ)に対応するエンコーダは、図1に示したエンコーダ、あるいは図6に示したエンコーダの場合と全く同様の構成であり、デコーダの構成のみが異なる。図11は、図6に示したエンコーダに対するデコーダを示している。なお、図1のエンコーダに対するデコーダは、その基本となる考え方および実現方法は以下に説明するように、図11の場合と同様である。従って、その図示および説明は省略する。

【0140】図11に示したデコーダの基本的な構成



は、図6に示したデコーダの場合とほぼ同様である。図6のデコーダと異なる点は、図11の復号器の逆可変長符号化器209によって復号されるフラグmismatchを利用せずに無視し、ミスマッチ制御器241によって常にDCT係数の奇数化によるミスマッチ制御を行なう点である。DCT係数のクリッピング器243は、このミスマッチ制御器241の後に置かれる。

【0141】上記のように構成する理由は以下の通りである。エンコーダ側とデコーダ側とで用いるミスマッチの対策方法が異なる場合、当然ながらミスマッチは起こり、一般にエンコードされたフレーム数が多くなるほどそのミスマッチは視覚的に感知されやすくなる。しかし、エンコーダ側でDCT係数の(7, 7)係数をtoggleするミスマッチ制御を行なって生成した符号化ビット列に対して、デコーダ側でDCT係数の奇数化によるミスマッチ制御を行なった場合、そのミスマッチの周波数成分はDCT係数の(7, 7)係数の成分だけにほぼ限定される。この成分は非常に高周波であるため、視覚的には比較的感知されにくい。

【0142】そのため、デコーダが行なうミスマッチの対策方法はDCT係数の奇数化によるミスマッチ制御のみと限定することによって、デコーダの演算あるいは回路を簡易化することにも良い。なお、このような簡略化を行なった場合、ミスマッチとして視覚的に感知されるべきノイズ成分を軽減あるいは除去するために、ノイズ軽減フィルタあるいはノイズ除去フィルタを用いても良い。このフィルタは、ミスマッチ制御の処理を行なった後や、画像を表示する装置にデコードした画像を送る直前等のように、主にノイズ成分を軽減、あるいは除去による効果が視覚的に現れる位置に置かれる。

【0143】図11に示した実施の形態における、より詳しい処理の流れ等について、以下説明する。

【0144】図12に示したフローチャートは、図11に示したデコーダにおける処理の流れと符号化ビット列の解読法の一例を示している。図11の実施の形態に対応するエンコーダは、図1あるいは図6のエンコーダの場合と全く同様の構成であり、デコーダの構成のみが異なる。図12のフローチャートは、図6のエンコーダに対する本実施の形態のデコーダのミスマッチ制御および逆量子化部分を示している。

【0145】なお、図1のエンコーダに対する本実施の形態のデコーダのミスマッチ制御および逆量子化部分は示していないが、その基本となる考え方および実現方法は以下に説明することと同様である。

【0146】復号された縦横8要素のDCT係数の各ブロックに対して、まずエンコーダ側で用いられた量子化行列の利用方法に従って、量子化行列を設定する。ステップS81においては、エンコーダによって符号化された符号化ビット列のうち、例えばVideo Object Layer Classのvideo\_object\_layer\_quant\_matrixのフラグが解

読される。そして、そのフラグがONである場合には、量子化行列を設定しない。あるいは、ステップS82において、設定器261により、量子化行列を設定した場合と量子化の演算を共通に行なえるように、全ての要素の値が(0以外でかつ正の値で)等しいなどの仮の量子化行列が設定される。そのフラグがOFFである場合には、ステップS83において、設定器232により、MPEG1/2型の量子化行列が設定される。ここで設定される量子化行列は、画像内符号化と画像間符号化とは異なる。

【0147】また、例えばVideo Object Layer Classのload\_intra\_quant\_mat、あるいはload\_nonintra\_quant\_matのフラグがONになっている場合には、あらかじめ用意されている量子化行列以外のものを設定するために、例えばintra\_quant\_mat[64]あるいはnonintra\_quant\_mat[64]として符号化された量子化行列の64個の要素を復号して用いる等の処理を行なわなければならない。

【0148】次に、エンコーダ側で用いられた量子化の方法に従って、逆量子化を行なう。ステップS84においては、エンコーダによって符号化された符号化ビット列のうち、例えばVideo Object Layer Classのvideo\_object\_layer\_quant\_typeのフラグが解読される。そのフラグがONである場合には、ステップS85に進み、逆量子化器223により、H.261/3型の逆量子化が行われる。一方、そのフラグがOFFである場合には、ステップS86に進み、逆量子化器224によりMPEG1/2型の逆量子化が行われる。逆量子化を行なう際には、量子化行列が設定されている場合にはそれを用い、量子化行列が設定されていない場合にはそれを用いない。あるいは、仮の量子化行列が設定されている場合には、それを用いる。

【0149】その後、ステップS87に進み、ミスマッチ制御器241により、DCT係数の奇数化によるミスマッチ制御が行われた後で、ステップS88において、クリッピング器243により、DCT係数のクリッピングが行われる。

【0150】図12に示した実施の形態における符号化シンタックスの基本的構成は、図9に示した実施の形態の場合と基本的に同様であるが、IDCTミスマッチの対策方法等が唯一つしかない点が大きく異なる。

【0151】復号された縦横8要素のDCT係数の各ブロックに対して、図11の実施の形態では、エンコーダ側で用いられたIDCTミスマッチの対策方法に依らず、常にDCT係数の奇数化によるミスマッチ制御を行なう。DCT係数のクリッピングは、このミスマッチ制御が行なわれた後で行なう。そのため、エンコーダによって符号化された符号化ビット列のうち、例えばVideo Object Layer Classのvideo\_object\_layer\_mismatchのフラグは利用されずに無視される。

【0152】上記に記した処理以外でも、処理の簡略化

による影響が大きくないものに対しては、逆可変長符号化器209によって復号されるフラグ等を利用せずに無視し、処理を簡略化しても良い。

【0153】以上説明したように、本発明によれば、既存の動画像の高効率符号化方法との互換性を取ることができる。また、量子化および逆量子化の方法、量子化行列の利用方法、IDCTミスマッチの対策方法等で、最適な方法の組合せを指定し、より高効率の符号化および復号を行うことができるため、符号化効率を向上させることができる。

【0154】なお、上記各フローチャートで示した処理を実行させるためのプログラムは、符号化装置および復号装置が備えるROM(read only memory)等に予め記憶させておいたり、CD-ROM(compact disc-read only memory)等に記録して使用者に供給するようにすることができる。

【0155】また、本発明の符号化装置および復号装置は、例えば、光磁気ディスクや磁気テープなどの記録媒体に記録し、それを再生してディスプレイ等に表示したり、テレビ会議システム、テレビ電話システム、放送用機器、マルチメディアデータベース検索システム等のように、動画像信号を伝送路を介して送信側から受信側に伝送し、受信側において、これを受信し、表示する場合や、動画像信号を編集し、記録する場合等にも応用することができる。

【0156】

【発明の効果】請求項1に記載の符号化装置、および請求項2に記載の符号化方法によれば、データを量子化する方法の任意のものと、量子化されたデータを逆量子化する方法の任意のものと、逆量子化されたデータに対してIDCTミスマッチ対策を施す方法の任意のものと、データを量子化するとき、および量子化されたデータを逆量子化するとき量子化行列を使用する任意の方法とを組み合わせることでデータを符号化する方法を制御するようにしたので、高効率の符号化が可能となる。

【0157】請求項3に記載の復号装置、および請求項4に記載の復号方法においては、データを逆量子化する方法の任意のものと、逆量子化されたデータに対してIDCTミスマッチ対策を施す方法の任意のものと、データを逆量子化するとき量子化行列を使用する任意の方法を組み合わせることでデータを復号する方法を制御するようにしたので、高効率の符号化が可能となる。

【0158】請求項5に記載の復号装置、および請求項6に記載の復号方法においては、データを逆量子化する方法の任意のものと、データを逆量子化するとき量子化行列を使用する任意の方法を組み合わせることでデータを復号する方法を制御するようにしたので、高効率の符号化が可能となる。

【図面の簡単な説明】

【図1】本発明の符号化装置を応用した符号化器の一実

施の形態の構成例を示すブロック図である。

【図2】本発明の復号装置を応用した復号器の一実施の形態の構成例を示すブロック図である。

【図3】図1の符号化器の量子化およびミスマッチ制御部分における処理手順と符号化ビット列の生成法を説明するフローチャートである。

【図4】図2の復号器のミスマッチ制御および逆量子化部分における処理手順と符号化ビット列の解読法を説明するフローチャートである。

10 【図5】図3および図4のVideo Object Layer Classにおけるシンタックスを示す図である。

【図6】本発明の符号化装置を応用した符号化器の他の実施の形態の構成例を示すブロック図である。

【図7】本発明の復号装置を応用した復号器の他の実施の形態の構成例を示すブロック図である。

【図8】図6の符号化器の量子化およびミスマッチ制御部分における処理手順と符号化ビット列の生成法を説明するフローチャートである。

20 【図9】図7の復号器のミスマッチ制御および逆量子化部分における処理手順と符号化ビット列の解読法を説明するフローチャートである。

【図10】図8および図9のVideo Object Layer Classにおけるシンタックスを示す図である。

【図11】本発明の復号装置を応用した復号器のさらに他の実施の形態の構成例を示すブロック図である。

【図12】図11の復号器のミスマッチ制御および逆量子化部分における処理手順と符号化ビット列の解読法を説明するフローチャートである。

30 【図13】従来の符号化器の一例の構成を示すブロック図である。

【図14】従来の復号器の一例の構成を示すブロック図である。

【図15】H. 261/3型の量子化の概念図である。

【図16】MPEG1/2型の量子化の概念図である。

【図17】MPEG1/2型の量子化行列の一例を示す図である。

【図18】符号化ビット列の構成例を示す図である。

【図19】VSのシンタックスを示す図である。

【図20】VOのシンタックスを示す図である。

40 【図21】Video Object Layer Classにおけるシンタックスを示す図である。

【図22】Video Object Layer Classにおけるシンタックスを示す図である。

【図23】Video Object Plane Classにおけるシンタックスを示す図である。

【図24】Video Object Plane Classにおけるシンタックスを示す図である。

【図25】Video Object Plane Classにおけるシンタックスを示す図である。

50 【図26】Video Object Plane Classにおけるシンタッ

クスを示す図である。

【図27】IおよびPピクチャ(VOP)におけるマクロブロックのシンタックスを示す図である。

【図28】Bピクチャ(VOP)でのマクロブロックのシンタックスを示す図である。

【図29】MODBの可変長符号を示す図である。

【図30】図13の符号化器の量子化およびミスマッチ制御部分における処理手順と符号化ビット列の生成法を説明するフローチャートである。

【図31】図14の復号器のミスマッチ制御および逆量子化部分における処理手順と符号化ビット列の読法を説明するフローチャートである。

【符号の説明】

101 動きベクトル検出器, 102 フレームメモ\*

\*り, 103, 203動き補償器, 104, 110, 210 加算器, 105 DCT器, 106 IDCT器, 109 可変長符号化器, 121 H. 261/3型の量子化器, 122 MPEG1/2型の量子化器, 123, 223 H. 261/3型の逆量子化器, 124, 125, 224, 225 MPEG1/2型の逆量子化器, 131 仮の量子化行列の設定器, 132, 232, 233MPEG1/2型の量子化行列の設定器, 141, 151, 241, 251係数を奇数化するミスマッチ制御器, 143, 152, 153, 243, 244, 252 DCT係数のクリッピング器, 154, 253 (7, 7)係数をtoggleするミスマッチ制御器, 209 逆可変長符号化器

【図5】

```

...
} lighting_change_in_sprite 1
}
video_object_layer_quant_type 2
if (video_object_layer_quant_type) {
  load_intra_quant_mat 1
  if (load_intra_quant_mat) {
    intra_quant_mat [64] 8*64
  }
  load_nonintra_quant_mat 1
  if (load_nonintra_quant_mat) {
    nonintra_quant_mat [64] 8*64
  }
  if (video_object_layer_shape == '10') {
    ...
  }
}
error_resilient_disable 1
...
}
( video_object_layer_quant_type = '01' : H.261/3 モード
  - '10' : MPEG1 モード
  - '11' : MPEG2 モード )

```

実施例1でのVideo Object Layer Class におけるシンタックス

【図10】

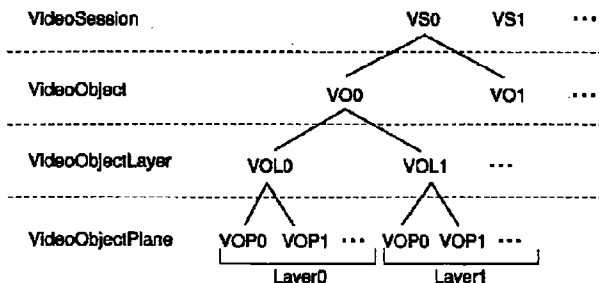
```

...
} lighting_change_in_sprite 1
}
video_object_layer_quant_matrix 1
if (video_object_layer_quant_matrix) {
  load_intra_quant_mat 1
  if (load_intra_quant_mat) {
    intra_quant_mat [64] 8*64
  }
  load_nonintra_quant_mat 1
  if (load_nonintra_quant_mat) {
    nonintra_quant_mat [64] 8*64
  }
  if (video_object_layer_shape == '10') {
    disable_gray_quant_update 1
    load_gray_intra_quant_mat 1
    if (load_gray_intra_quant_mat) {
      gray_intra_quant_mat [64] 8*64
    }
    load_gray_nonintra_quant_mat 1
    if (load_gray_nonintra_quant_mat) {
      gray_nonintra_quant_mat [64] 8*64
    }
  }
}
video_object_layer_quant_type 1
video_object_layer_mismatch 1
error_resilient_disable 1
...
}

```

Video Object Layer Class におけるシンタックス

【図18】



符号化ビット列の構成

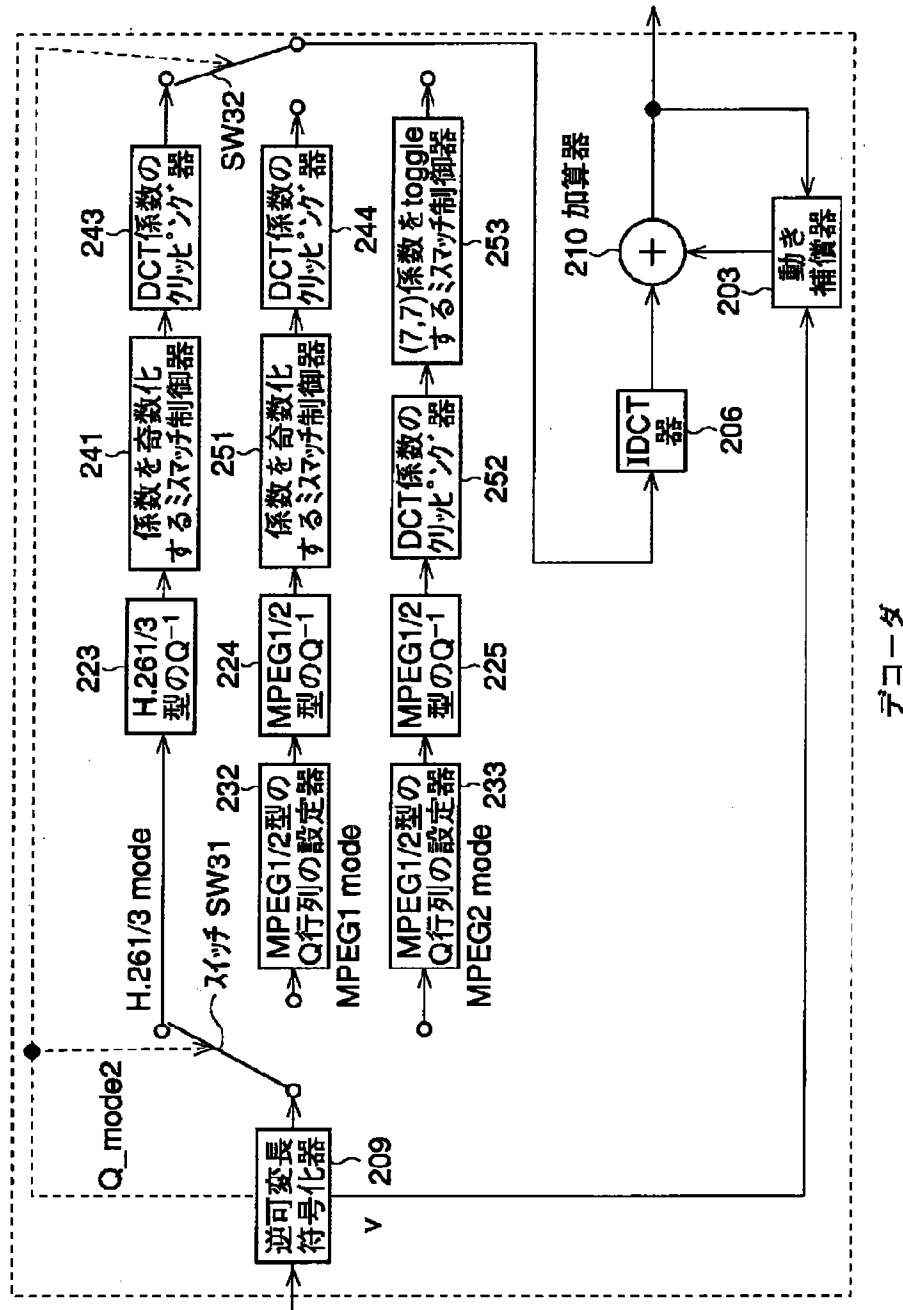
【図19】

Syntax	No. of bits	Minim. -onic
VideoSession () { video_session_start_code do { VideoObject () } while (nextbits_bytestigned () == video_object_start_code) next_start_code () video_session_end_code }	sc+8-32	sc+8-32

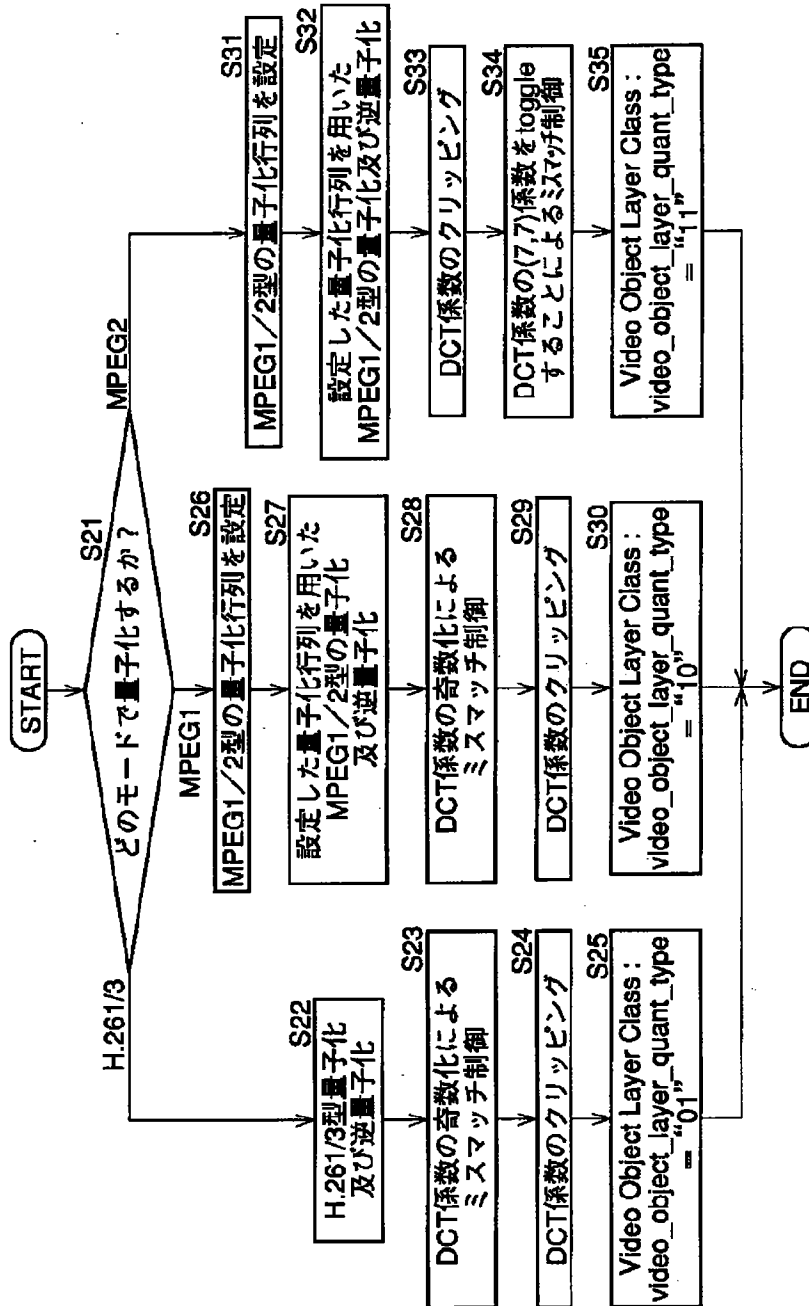
VSのシンタックス

The diagram illustrates a video encoding process. It starts with an input signal that splits into two paths. One path goes through a 7-tap filter (102) and an adder (104) to produce a motion-compensated signal (101). The other path goes through a DCT block (105) and a switch (SW1). The output of SW1 is then processed by an H.261/3 mode Q-row setting block (132), which branches into H.261/3 mode Q (121) and MPEG1/2 mode Q (122). These signals are then processed by MPEG1/2 type Q-1 blocks (151, 153) and H.261/3 mode Q-1 blocks (141, 143). The outputs of these blocks are then processed by DCT coefficient quantization blocks (154, 155) and (7,7) coefficient toggle blocks (156, 157). The final outputs are then processed by IDCT blocks (106, 107) and a motion compensation block (108) to produce the final output signal.

【図2】

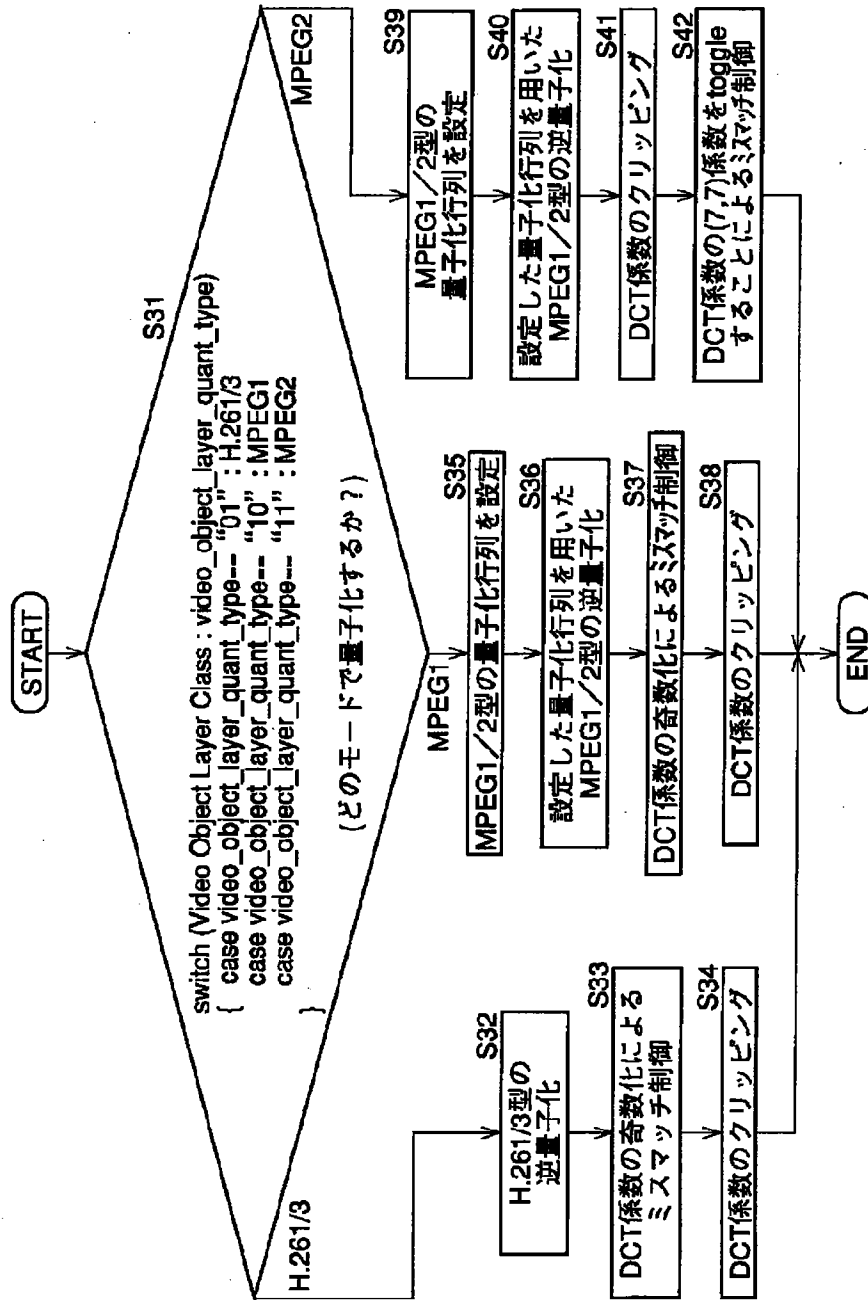


【図3】



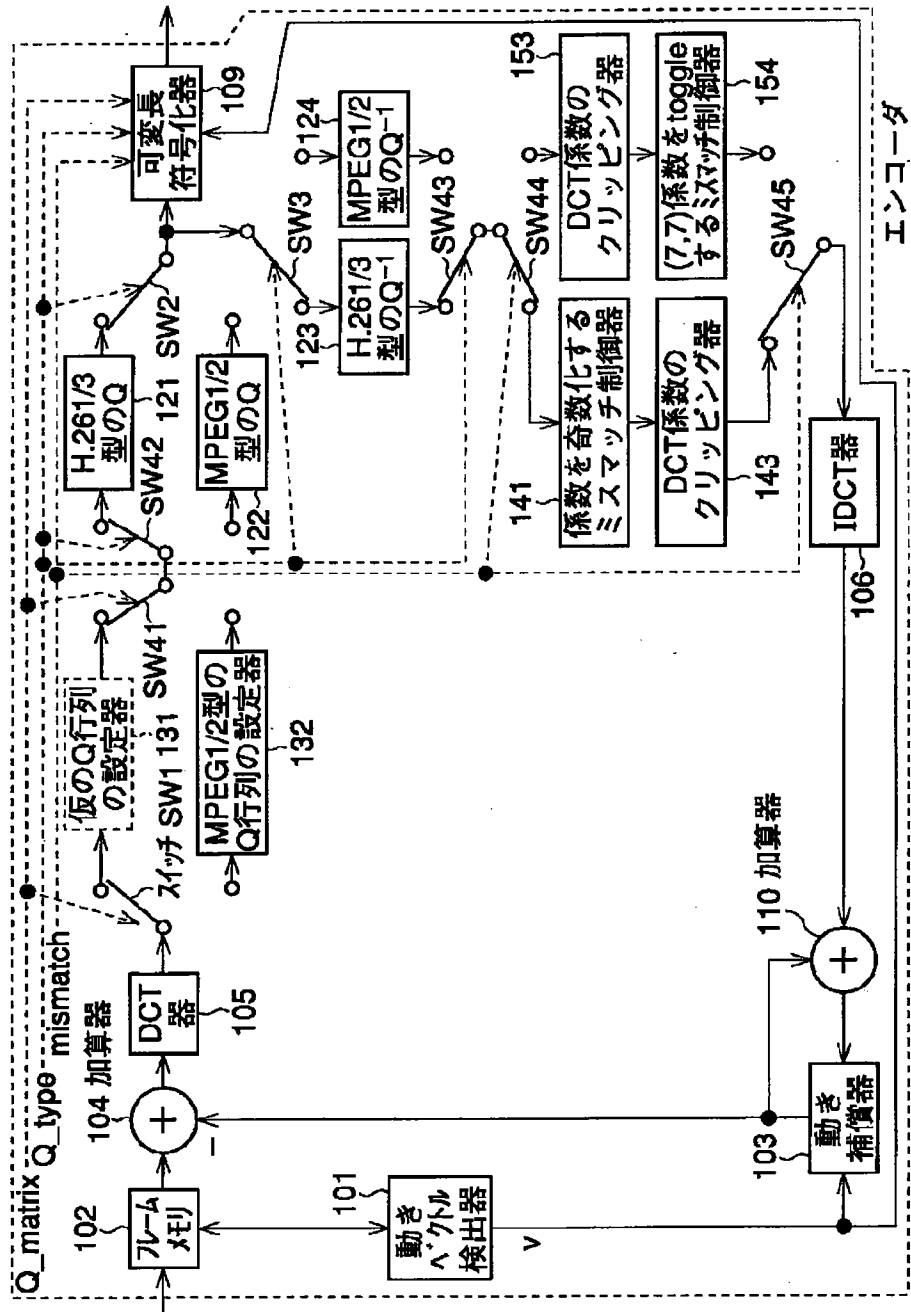
エンコーダの量子化及びミスマッチ制御部分における  
処理の流れと符号化ビット列の生成法

【図4】



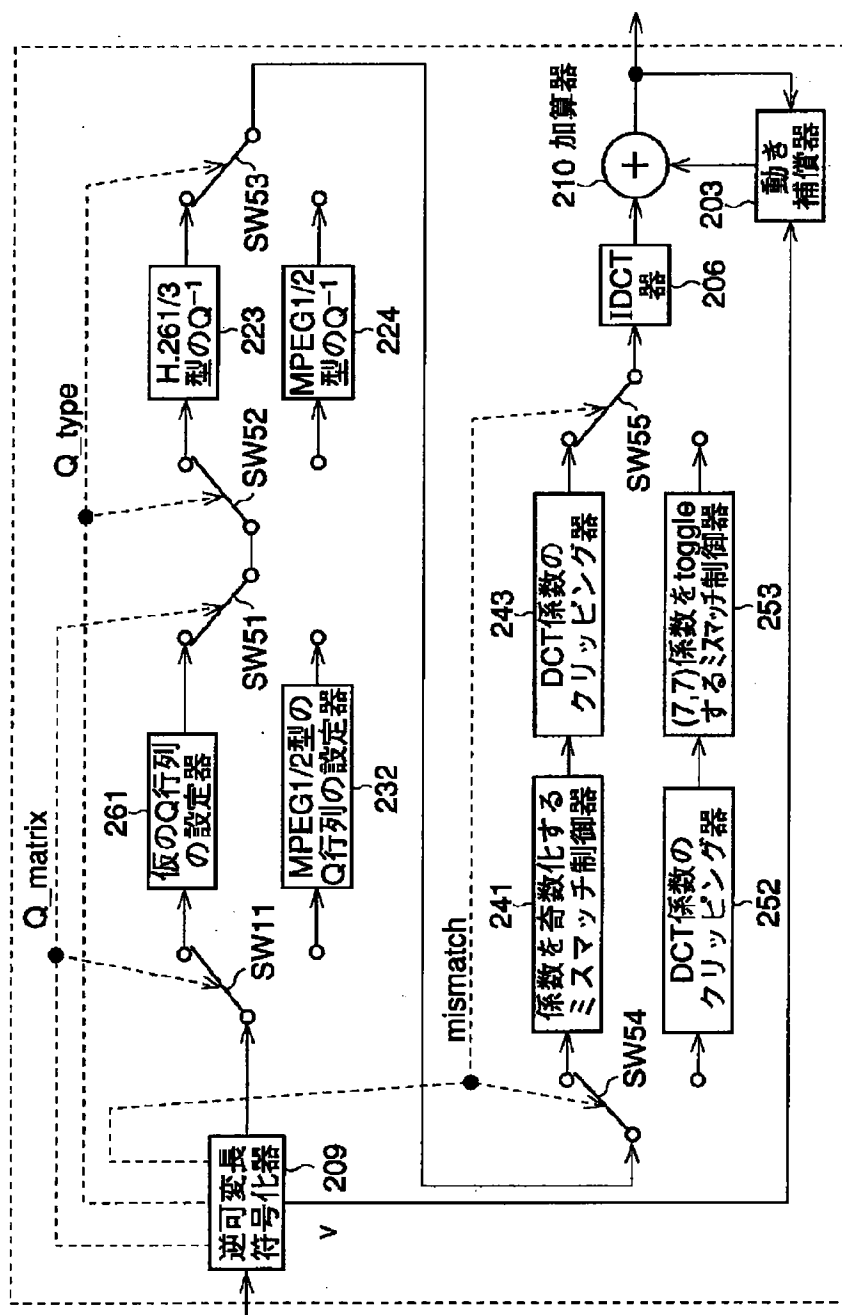
デコーダのミスマッチ制御及び逆量子化部分における処理の流れと符号化ビット列の解読法

【図 6】

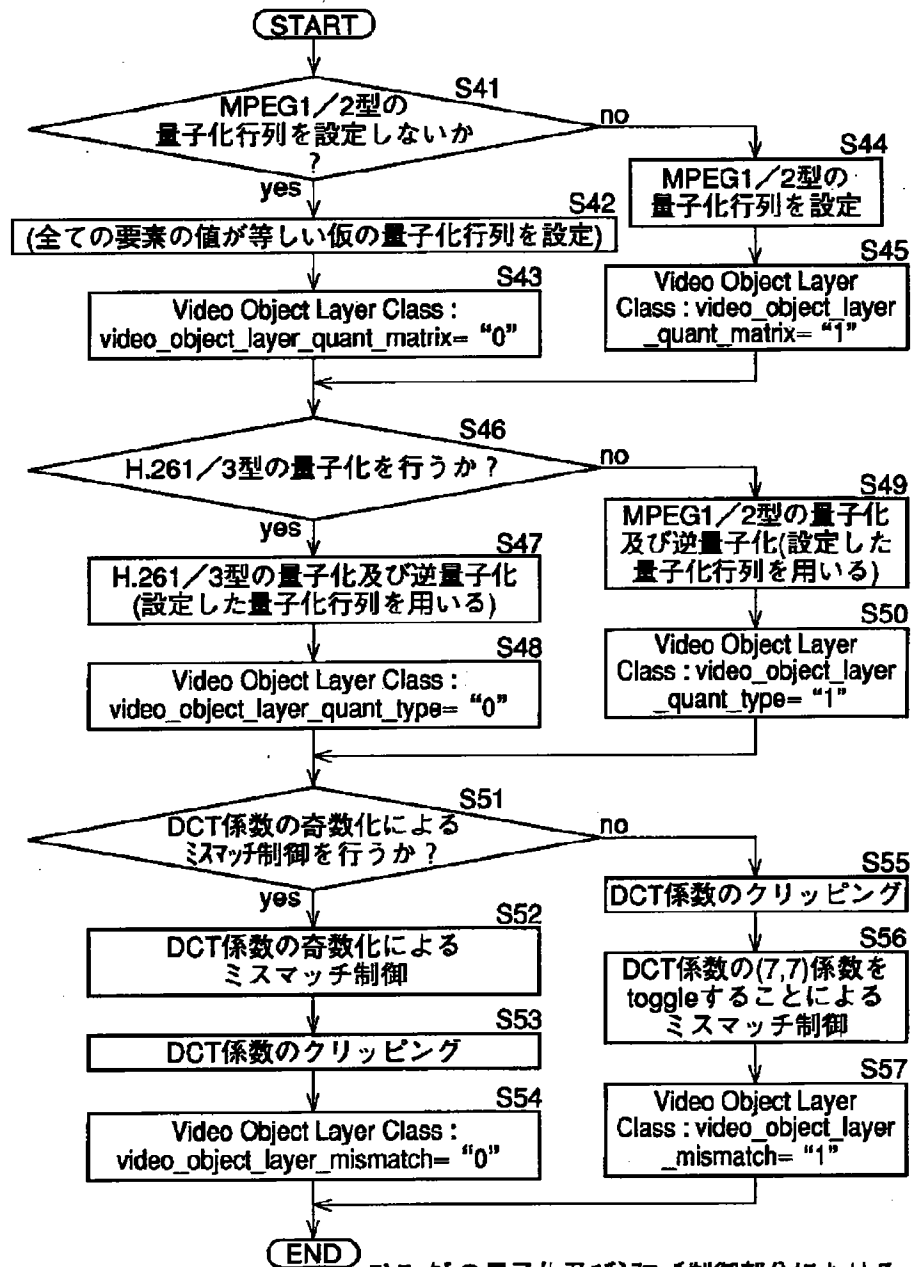




テコ一タ

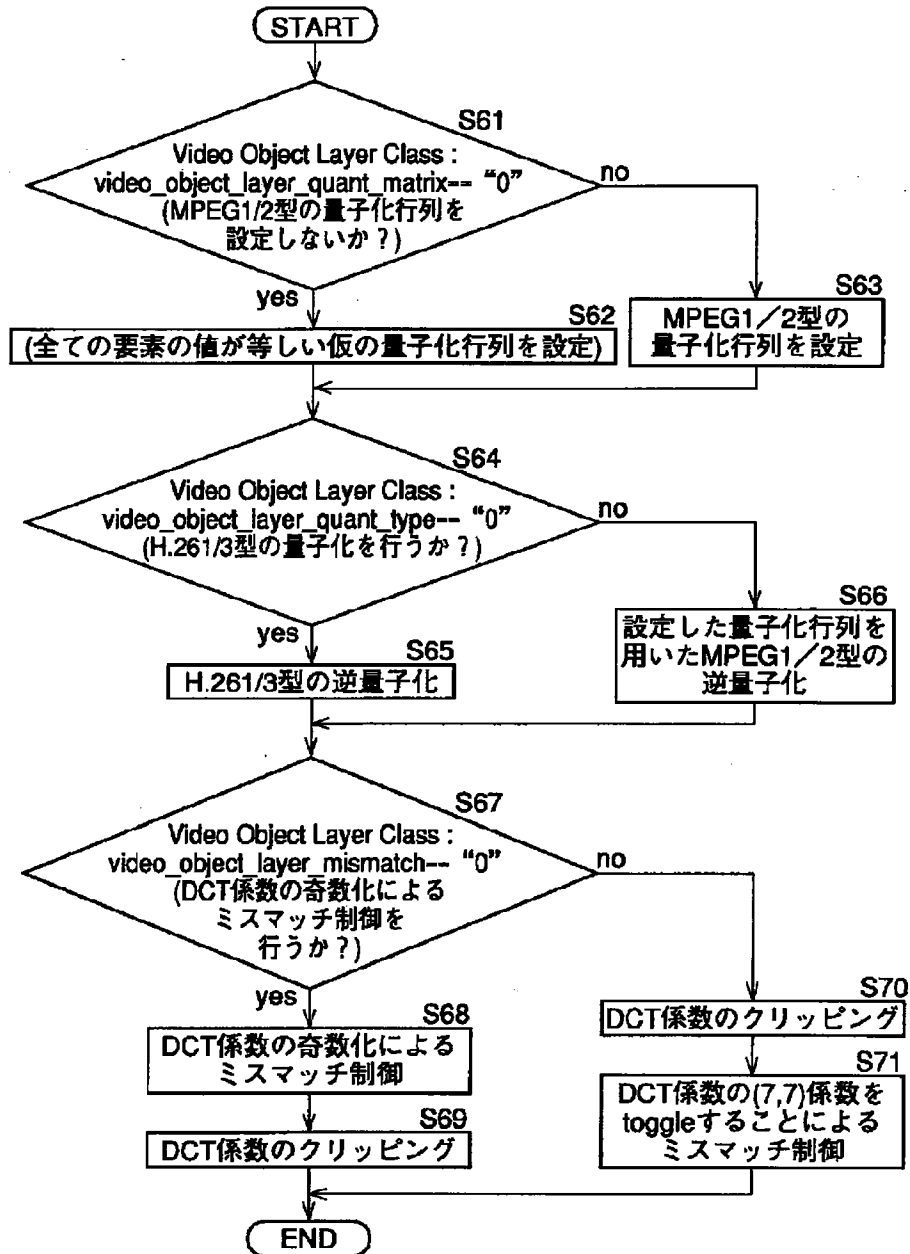


【図8】



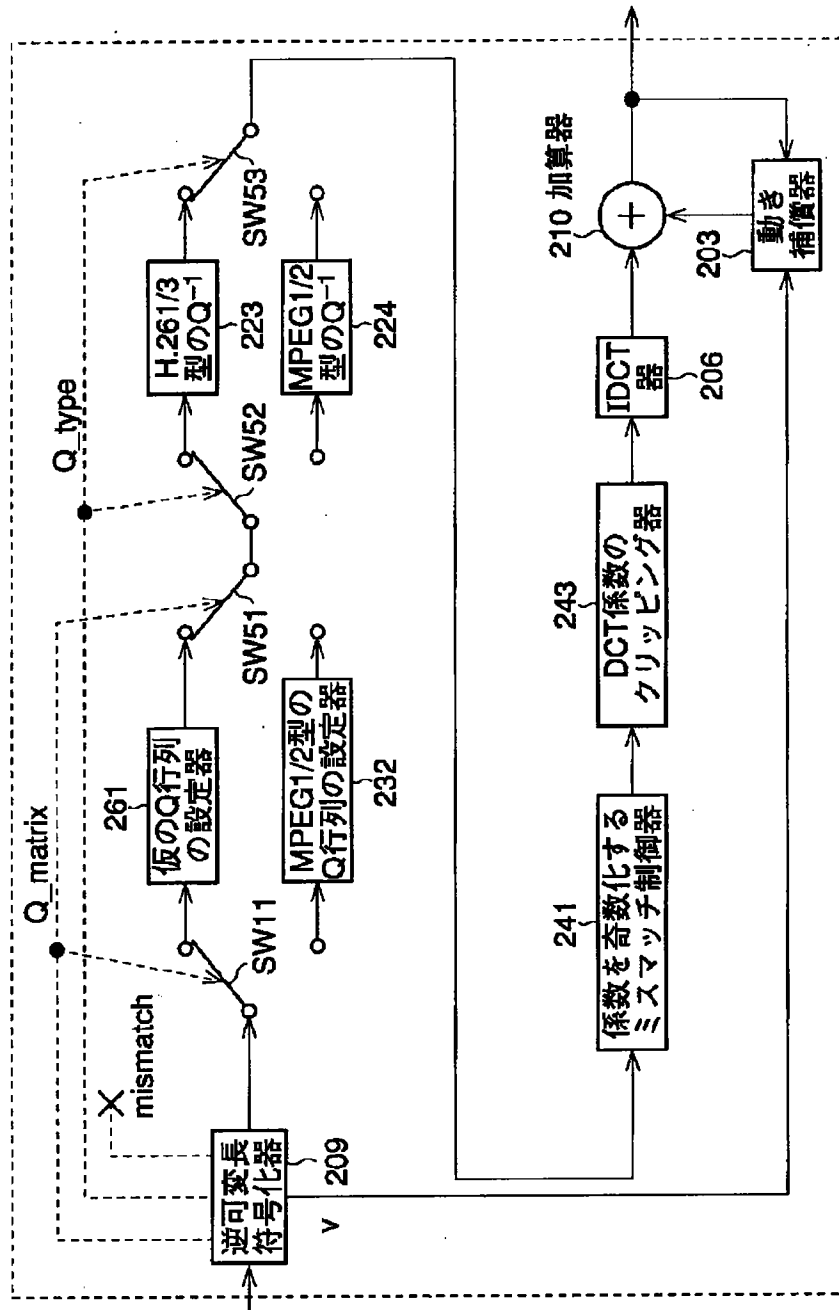
エンコーダの量子化及びミスマッチ制御部分における  
処理の流れと符号化ビット列の生成法

【図9】

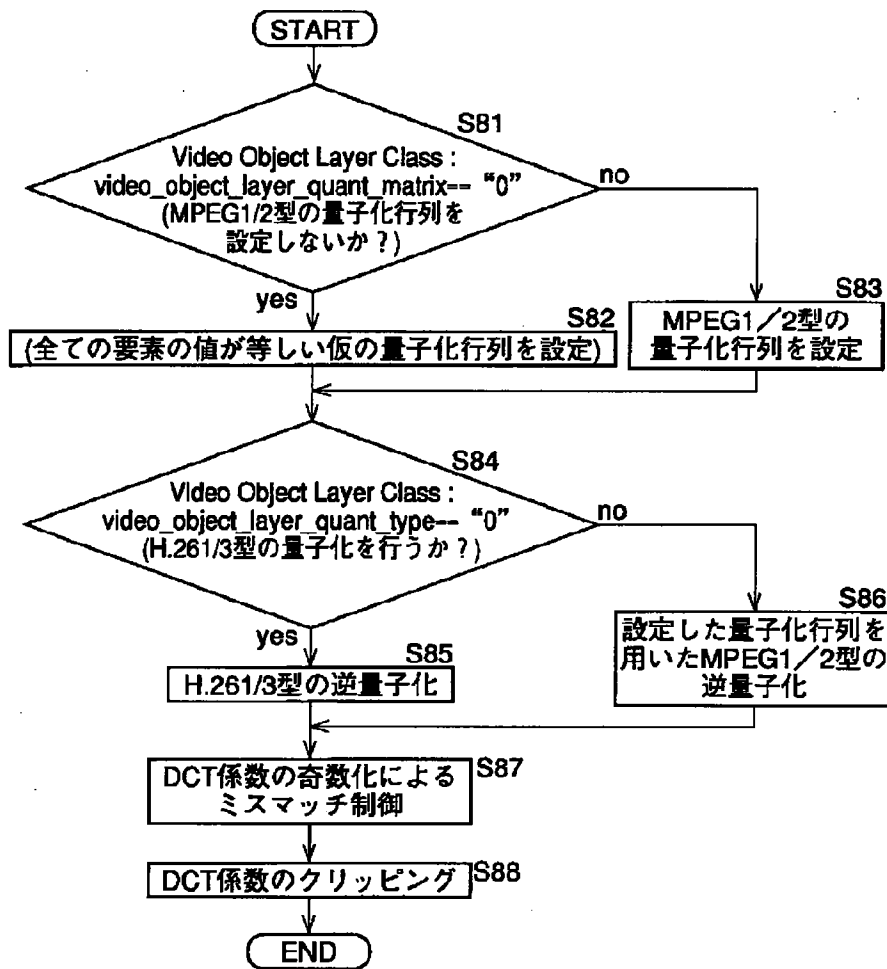


デコーダのミスマッチ制御及び逆量子化部分における処理の流れと  
符号化ビット列の解釈法

データ



【図12】



デコーダのミスマッチ制御及び逆量子化部分における処理の流れと  
符号化ビット列の解釈法

【図20】

Syntax	No. of bits	Mnemonic
VideoObject () { video_object_start_code video_object_id do { VideoObjectLayer () } while (nextbits_bytealigned () == video_object_layer_start_code) do { StillImageObjectLayer () } while (nextbits_bytealigned () == stillimage_object_layer_start_code) next_start_code () }	so+8-27 5	-enc

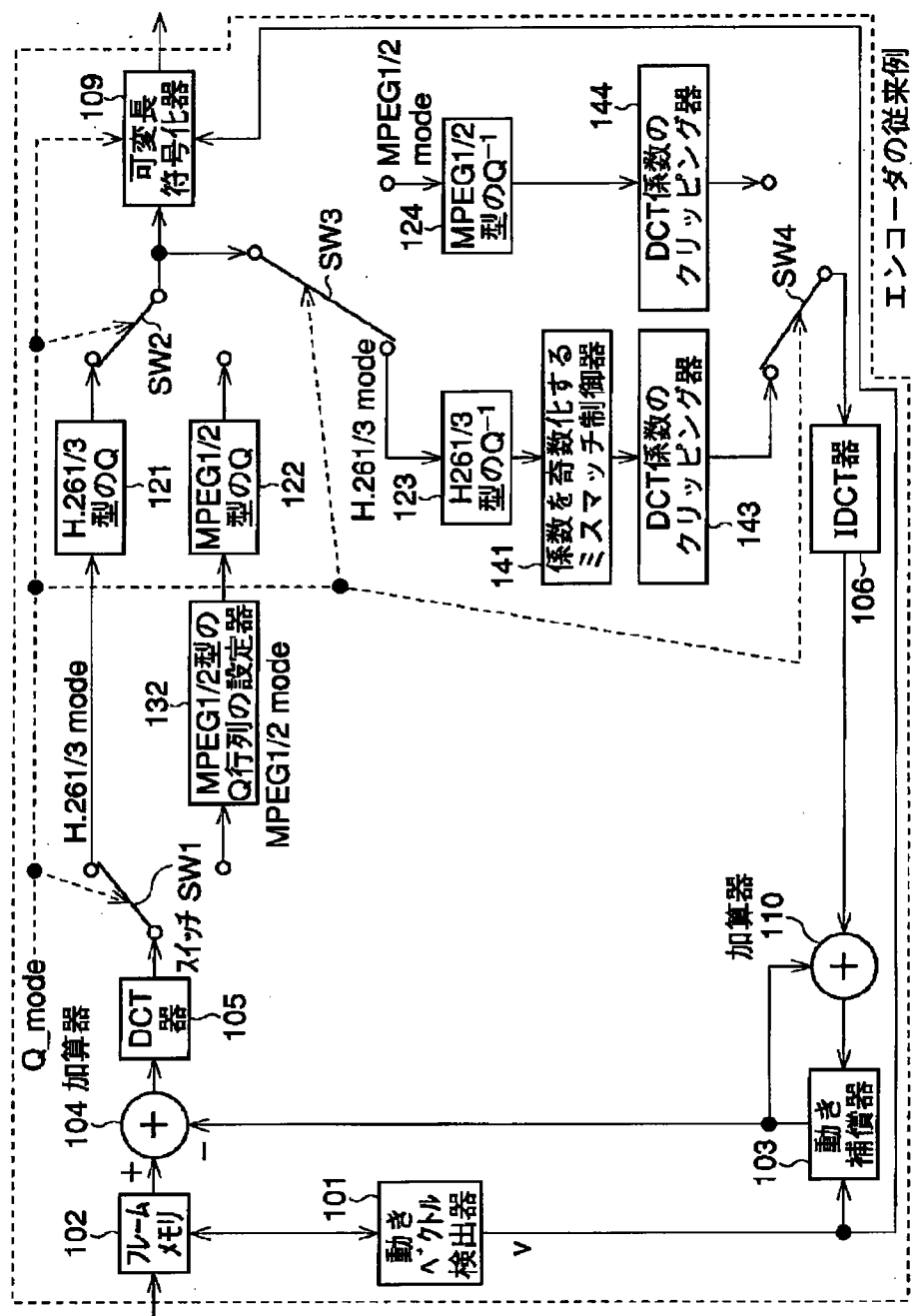
VOのシンタックス

【図29】

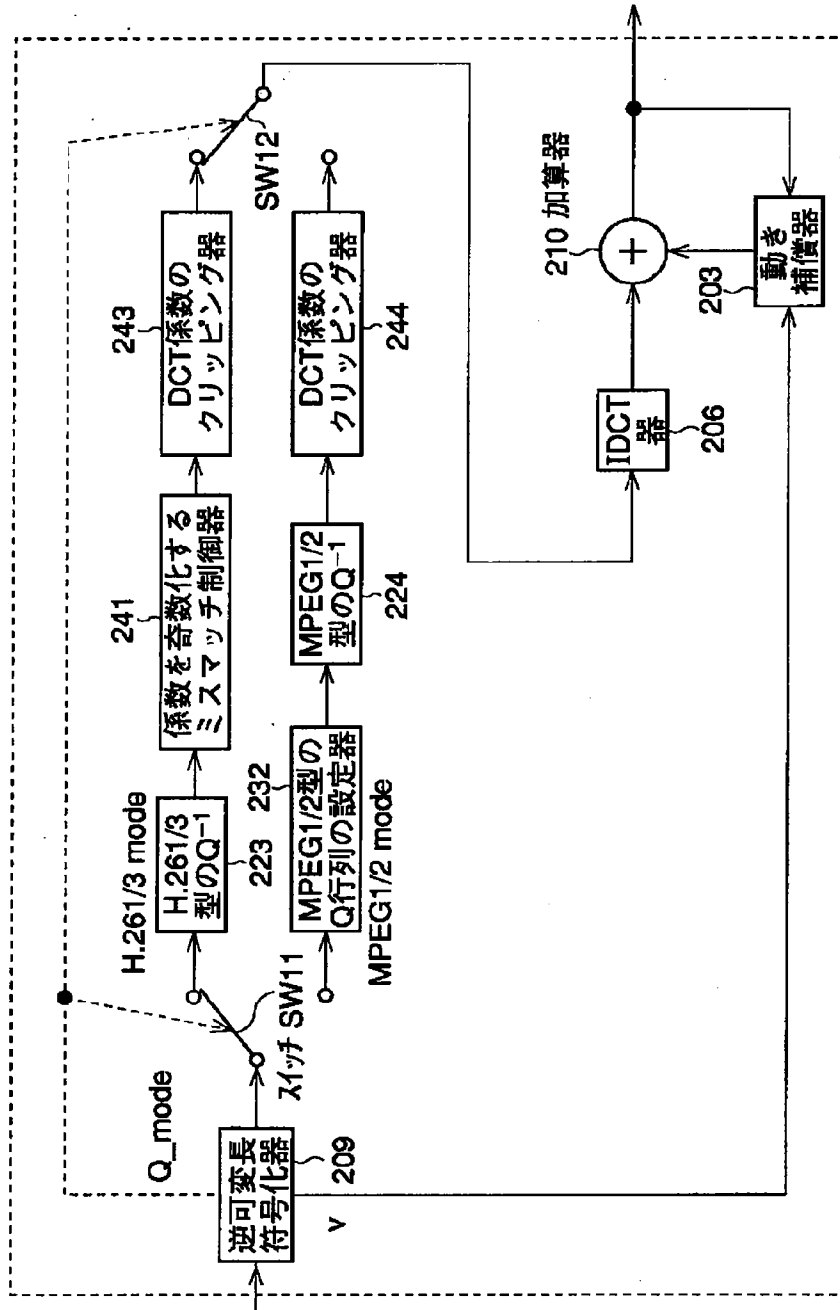
Index	CBPB	MBTYPE	Number of bits	code
0			1	0
1		x	2	10
2	x	x	2	11

MODBの可変長符号

【図13】

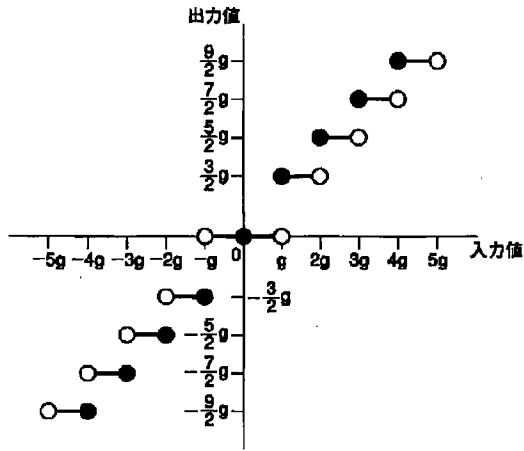


【図14】



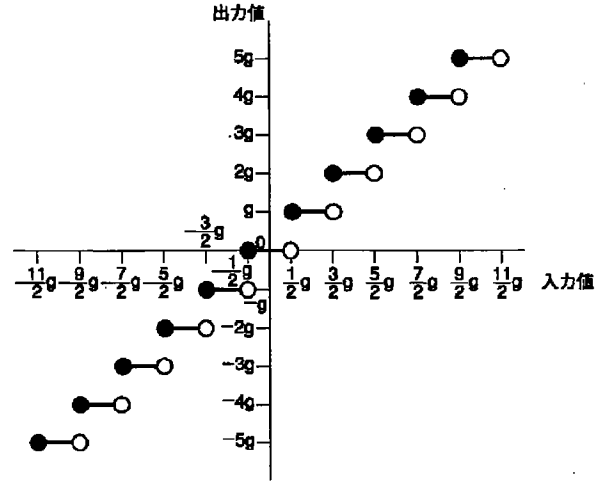
デコーダの従来例

【図15】



H.261/3型の量子化の概念図

【図16】



MPEG1/2型の量子化の概念図

【図17】

8	17	18	19	21	23	25	27
17	18	19	21	23	25	27	28
20	21	22	23	24	26	28	30
21	22	23	24	26	28	30	32
22	23	24	26	28	30	32	35
23	24	26	28	30	32	35	38
25	26	28	30	32	35	38	41
27	28	30	32	35	38	41	45

(A) 画像内符号化

16	17	18	19	20	21	22	23
17	18	19	20	21	22	23	24
18	19	20	21	22	23	24	25
19	20	21	22	23	24	26	27
20	21	22	23	25	26	27	28
21	22	23	24	26	27	28	30
22	23	24	26	27	28	30	31
23	24	25	27	28	30	31	33

(B) 画像間符号化

【図21】

21-1		No. of bits	Mnem
Syntax			-only
VideoObjectLayer () {			
video_object_layer_start_code		sc+4=28	
video_object_layer_id		4	
video_object_layer_shape		2	
if (video_object_layer_shape == '00') {			
video_object_layer_width		10	
video_object_layer_height		10	
}			
video_object_layer_shape_effects		4	
if (video_object_layer_shape_effects == '0001' )			
(video_object_layer_shape_effects == '0011' )			
(video_object_layer_shape_effects == '0100' )			
(video_object_layer_shape_effects == '0101' )			
video_object_layer_feather_dist		3	
if (video_object_layer_shape_effects == '0100' )			
(video_object_layer_shape_effects == '0101' ) {			
for (i=0; i<video_object_layer_feather_dist; i++)			
feathering_filter ();		8*15	
}			
video_object_layer_sprite_usage		2	
if (video_object_layer_sprite_usage !=			
SPRITE_NOT_USED) {			
if (video_object_layer_sprite_usage ==			
ON-LINE_SPRITE) {			
sprite_hdim		13	
sprite_vdim		13	
no_of_sprites		3	
}			
no_of_sprite_points		6	
if (video_object_layer_sprite_usage != GMC) {			
for (i=0; i<no_of_sprite_points; i++) {			
sprite_point[i].x_coordinate		13	
sprite_point[i].y_coordinate		13	
}			
for (i=0; i<no_of_sprite_points; i++) {			
sprite_point[i].x_coordinate		13	
sprite_point[i].y_coordinate		13	
}			
lighting_change_in_sprite		1	
}			
video_object_layer_quant_type		1	
if (video_object_layer_quant_type) {			
load_intra_quant_mat		1	
if (load_intra_quant_mat)			



【図22】

Syntax	No. of bits	Mnem -onic
Intra_quant_mat [64]	8*64	
load_nonintra_quant_mat	1	
If (load_nonintra_quant_mat)		
nonintra_quant_mat [64]	8*64	
If (video_object_layer_shape == "10" ) {		
disable_gray_quant_update	1	
load_gray_intra_quant_mat	1	
If (load_gray_intra_quant_mat)		
gray_intra_quant_mat [64]	8*64	
load_gray_nonintra_quant_mat	1	
If (load_gray_nonintra_quant_mat)		
gray_nonintra_quant_mat [64]	8*64	
}		
error_resilient_disable	1	
If (!error_resilient_disable)		
reversible_VLC	1	
intra_acdc_pred_disable	1	
video_object_layer_code_forward	2	
video_object_layer_code_backward	2	
separate_motion_shape_texture	1	
If (video_object_layer_sprite_usage ==		
STATIC_SPRITE)		
{		
encodeSpritePiece ()		
}		
Scalability	1	
If (scalability) {		
ref_layer_id	4	
ref_layer_sampling_dirac	1	
hor_sampling_factor_n	5	
hor_sampling_factor_m	5	
vert_sampling_factor_n	5	
vert_sampling_factor_m	5	
enhancement_type	1	
}		
do {		
VideoObjectPlane ()		
} while (nextbits_bytealigned ()==		
video_object_plane_start_code)		
next_start_code ()		

【図23】

Syntax	No. of bits	Mnem -onic
VideoObjectPlane () {		
VOP_start_code	sc+8-32	
do {		
module_time_base	1	
} while (module_time_base != "0" )		
VOP_time_increment	10	
VOP_prediction_type	2	
If ( ( ( video_object_layer_sprite_usage		
SPRITE_NOT_USED ) &&		
(VOP_prediction_type == SPRITE) ) {		
If (no_of_sprite_points>0) {		
encode VOP_points ()		
} If (lighting_change_in_sprites) {		
lighting_change_factor_encode ()		
} If (video_object_layer_sprite_usage ==		
STATIC_SPRITE) {		
If (sprite_transmit_mode != STOP) {		
do {		
sprite_transmit_mode = checkSpriteTransMode ()		
sprite_transmit_mode	2	
If (sprite_transmit_mode != PAUSE)		
If (sprite_transmit_mode == NEXT) {		
If (no_sprite_transmitted == no_of_sprites)		
Sprite_transmit_mode = STOP		
else {		
no_sprite_transmitted = no_sprite_transmitted +1		
encodeSpritePiece ()		
}		
} else		
encodeSpritePiece ()		
} while ((sprite_transmit_mode		
STOP) && (sprite_transmit_mode != PAUSE))		
return ()		
} else If (video_object_layer_sprite_usage == ON-		
LINE_SPRITE) {		
blending_factor	8	
}		
} If (video_object_layer_shape != "00" ) {		
VOP_width	10	
VOP_height	10	

【図26】

Syntax	No. of bits	Mnem -onic
else {		
If (background_composition) {		
load_backward_shape	1	
If (load_backward_shape) {		
backward_shape_width	10	
backward_shape_height	10	
backward_shape_horizontal_mc_spatial_ref	10	
marker_bit	1	
backward_shape_vertical_mc_spatial_ref	10	
backward_shape_coding ()		
load_forward_shape	1	
If (load_forward_shape)		
forward_shape_width	10	
forward_shape_height	10	
forward_shape_horizontal_mc_spatial_ref	10	
marker_bit	1	
forward_shape_vertical_mc_spatial_ref	10	
forward_shape_coding ()		
}		
ref_select_code	2	
If (VOP_prediction_type == "01"    VOP_prediction_type ==		
"10" ) {		
forward_temporal_ref	10	
If (VOP_prediction_type == "10" ) {		
marker_bit	1	
backward_temporal_ref	10	
}		
} combined_motion_shape_texture_coding ()		
next_start_code ()		

【図24】

Syntax	No. of bits	Mnem -onic
VOP_horizontal_mc_spatial_ref	10	
marker_bit	1	
VOP_vertical_mc_spatial_ref	10	
if (scalability && enhancement_type)		
background_composition	1	
}		
disable_sadct	1	
if (VOP_prediction_type == "10")		
VOP_quant	2	
else {		
VOP_quant	5	
if (video_object_layer_shape == "10")		
VOP_qray_quant	6	
}		
if ((video_object_layer_shape_effects == "0010")		
(video_object_layer_shape_effects == "0011")		
(video_object_layer_shape_effects == "0101")) {		
VOP_constant_alpha	1	
if (VOP_constant_alpha)		
VOP_constant_alpha_value	8	
}		
if (!scalability) {		
if (separate_motion_shape_texture)		
if (error_resilience_disable)		
combined_motion_shape_texture_coding ()		
else		
do {		
combined_motion_shape_texture_coding ()		
} while (nextbits_bytaligned () != 0000 0000		
0000 0000)		
if (nextbits_bytaligned () != 000 0000 0000		
0000 0000 0000) {		
next_resync_marker ()	17	
resync_marker		
macroblock_number	1-12	
quant_scale	5	
header_extension_code	1	
if (header_extension_code == 1) {		
do {		
modulo_time_base	1	
} while (modulo_time_base != 0)		
VOP_time_increment	10	
}		
}		

【図25】

Syntax	No. of bits	Mnem -onic
} while (nextbits_bytaligned () != 000 0000 0000		
0000 0000 0000)		
else {		
if (video_object_layer_shape != "00") {		
do {		
first_shape_code	1-3	
} while (count of macroblocks != total		
number of macroblocks)		
if (error_resilience_disable)		
{		
motion_coding ()		
if (video_object_layer_shape != "00")		
shape_coding ()		
texture_coding ()		
}		
else		
do {		
motion_coding ()		
} while (next bits () != '1010 0000 0000 0000 1')	17	
motion_marker		
if (video_object_layer_shape != "00")		
shape_coding ()		
do {		
texture_coding ()		
} while (nextbits_bytaligned () != '0000 0000		
0000 0000')		
if (nextbits_bytaligned () != 000 0000 0000		
0000 0000 0000) {		
next_resync_marker ()	17	
resync_marker		
macroblock_number	1-12	
quant_scale	5	
header_extension_code	1	
if (header_extension_code == 1) {		
do {		
modulo_time_base	1	
} while (modulo_time_base != 0)		
VOP_time_increment	10	
}		
} while (nextbits_bytaligned () != 000 0000 0000		
0000 0000 0000)		
}		

【図27】

First_shape_code	COD	MCBPC				
Second_shape_code	MVD_shape	ST	VLC_binary/Vpass	RLB/ULB		
AC_pred_flag	CBPY	DQUANT	MVD	MVD2	MVD3	MVD4
CODA	Alpha_ACpred_flag	CBPA	Alpha Block Data	Block Data		

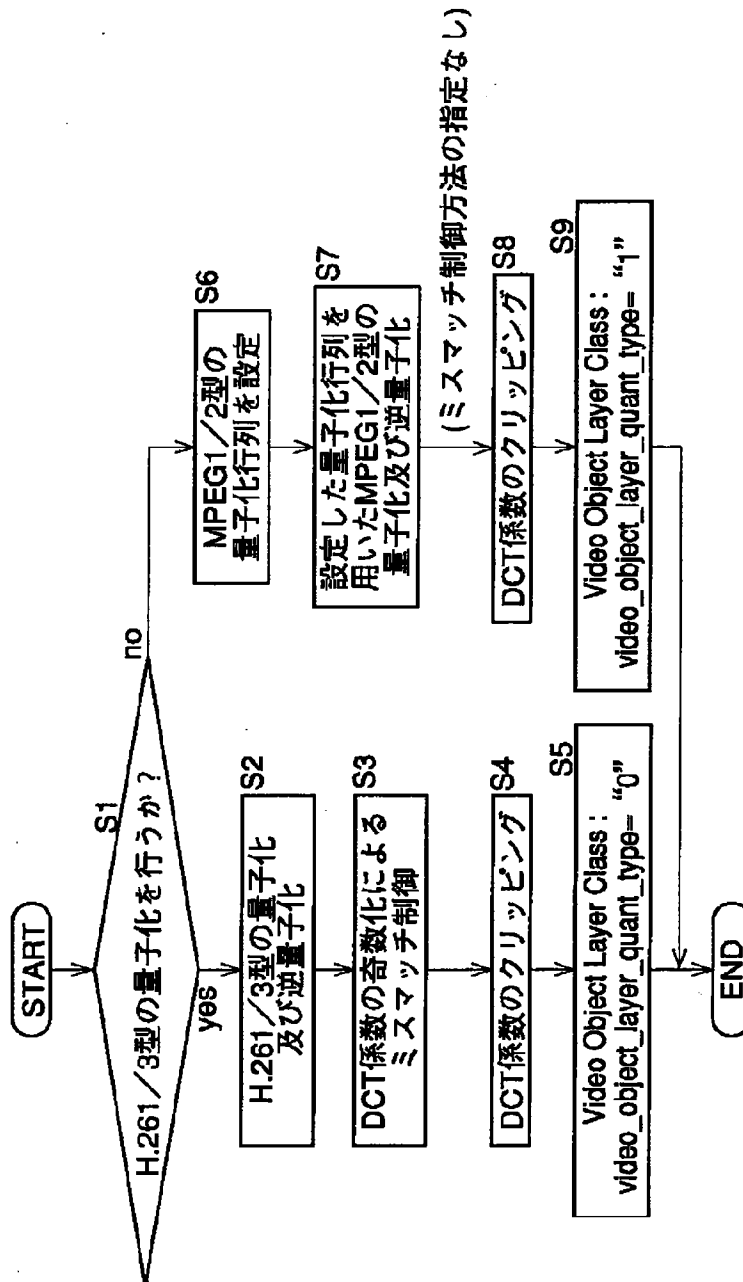
IおよびPピクチャ(VOP)におけるマクロブロックのシンタックス

【図28】

First_shape_code	MODB	MBTYPE				
Second_shape_code	MVD_shape	CR	ST	VLC_binary/Vpass	RLB/ULB	
CBPB	DQUANT	MVDi	MVDb	MVDB		
CODA	CBPBA	Alpha Block Data	Block Data			

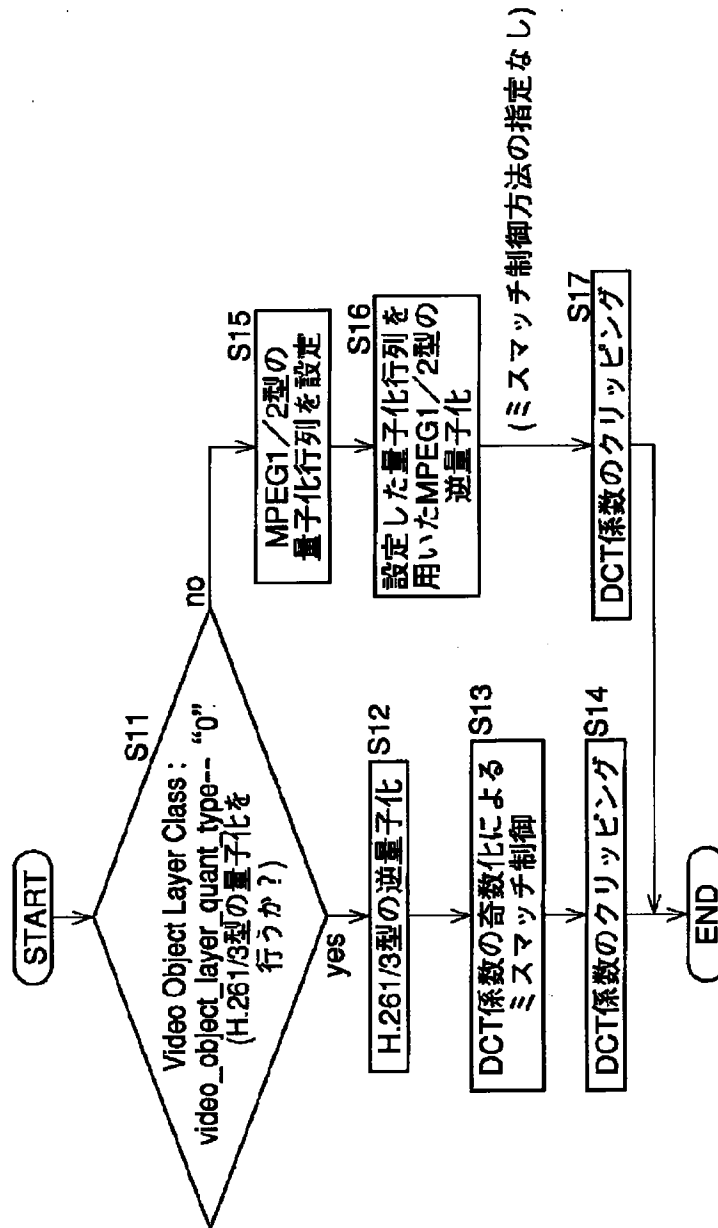
Bピクチャ(VOP)でのマクロブロックのシンタックス

【図30】



エンコーダの量子化及びミスマッチ制御部分における処理の流れと  
符号化ビット列の生成法、従来例

【図31】



デコーダのミスマッチ制御及び逆量子化部分における処理の流れと  
符号化ビット列の解読法、従来例